NAVAL RESEARCH LOGISTICS QUARTERLY

DECEMBER 1979
VOL. 26, NO. 4

OFFICE OF NAVAL RESEARCH

NAVSO P-1278

NAVAL RESEARCH LOGISTICS QUARTERLY

The Naval Research Logistics Quarterly is devoted to the dissemination of scientific information in logistics a will publish research and expository papers, including those in certain areas of mathematics, statistics, and economi relevant to the over-all effort to improve the efficiency and effectiveness of logistics operations.

Information for Contributors is indicated on inside back cover.

The Naval Research Logistics Quarterly is published by the Office of Naval Research in the months of March, Ju September, and December and can be purchased from the Superintendent of Documents, U.S. Government Printi Office, Washington, D.C. 20402. Subscription Price: $11.15 a year in the U.S. and Canada, $13.95 elsewhere. Cost individual issues may be obtained from the Superintendent of Documents.

Issuance of this periodical approved in accordance with Department of the Navy Publications and Printing Regulatio P-35 (Revised 1-74).

OPTIMAL SET PARTITIONING, MATCHINGS AND LAGRANGIAN DUALITY*

George L. Nemhauser

School of Operations Research
and Industrial Engineering
Cornell University
Ithaca, New York

Glenn M. Weber

Christopher Newport College
Newport News, Virginia

ABSTRACT

We formulate the set partitioning problem as a matching problem with simple side constraints. As a result we obtain a Lagrangian relaxation of the set partitioning problem in which the primal problem is a matching problem. To solve the Lagrangian dual we must solve a sequence of matching problems each with different edge-weights. We use the cyclic coordinate method to iterate the multipliers, which implies that successive matching problems differ in only two edge-weights. This enables us to use sensitivity analysis to modify one optimal matching to obtain the next one. We give theoretical and empirical comparisons of these dual bounds with the conventional linear programming ones.

1. INTRODUCTION

We consider the set partitioning problem

$$\max \sum_{j=1}^{n} d_j y_j$$

$$(SP) \qquad \sum_{j=1}^{n} a_{ij} y_j = 1, \ i = 1, \ \ldots, \ m$$

$$y_j \in \{0,1\}, \ j = 1, \ \ldots, \ n$$

where d_j is an arbitrary real number for all j and $a_{ij} \in \{0,1\}$ for all i and j. Balas and Padberg [1] give a survey of applications and methods for solving the set partitioning problem. Except for algorithms developed for small size problems, most algorithms for solving the set partitioning problem use linear programming relaxations. However, for large size problems, because of degeneracy, the linear programs obtained by replacing the binary restriction on each y_j in (SP) by $y_j \geqslant 0$ often are difficult to solve (Marsten [12]). As a result, the typically large size (and sparse) set partitioning problem sometimes cannot be solved.

*This work has been supported by National Science Foundation Grant ENG75-00568 to Cornell University.

We consider a different relaxation that uses matchings on graphs and Lagrangian duality. This is accomplished by reformulating the set partitioning problem as a (weighted, perfect) matching problem, a version of (SP) in which $\sum_{i=1}^{m} a_{ij} = 2$ for $j = 1, \ldots, n$, with simple side constraints. The side constraints are incorporated into the objective function in a Lagrangian fashion, resulting in the primal Lagrangian matching relaxation. The matching problem is one of the few tractable combinatorial problems, and thus is an attractive relaxation for large set partitioning problems.

2. LAGRANGIAN MATCHING RELAXATION

In (SP) let $a_j = (a_{1j}, \ldots, a_{mj})$ and suppose that for all j, $\sum_{i=1}^{m} a_{ij} = 2K_j$, for some integer K_j. This places no limitation on the generality of (SP) since if $\sum_{i=1}^{m} a_{ij} = 2K_j - 1$ then a new constraint $y_j \leqslant 1$ can be added to the problem. We replace a_j by the set of columns $\{a_j^k\}_{k=1}^{K_j}$, where $\sum_{k=1}^{K_j} a_j^k = a_j$, $\sum_{i=1}^{m} a_{ij}^k = 2$, $k = 1, \ldots, K_j$ and $a_{ij}^k \in \{0,1\}$, for all i and k. One can form these columns in such a way that both nonzero components of a_j^k precede those of a_j^{k+1}, for all k. Column a_j^k is given an objective function coefficient of $c_j = d_j/K_j$ and is associated with the variable $x_{jk} \in \{0,1\}$. The $\{x_{jk}\}$ are required to satisfy $x_{jk} = x_{j,k+1}$, $k = 1, \ldots, K_j - 1$, for all j. We thus obtain a problem equivalent to (SP) given by

$$
\begin{aligned}
&\max \sum_{j=1}^{n} \sum_{k=1}^{K_j} c_j x_{jk} \\
(MS) \qquad &\sum_{j=1}^{n} \sum_{k=1}^{K_j} a_{ij}^k x_{jk} = 1, \quad i = 1, \ldots, m \\
&x_{jk} - x_{j,k+1} = 0, \quad j = 1, \ldots, n \text{ and } k = 1, \ldots, K_j - 1 \\
&x_{jk} \in \{0,1\}, \text{ all } j \text{ and } k,
\end{aligned}
$$

which is a matching problem with side constraints $x_{jk} = x_{j,k+1}$. Using matrix notation, this problem can be written as

$$
\begin{aligned}
&\max cx \\
&Mx = 1 \\
&Sx = 0 \\
&x \text{ binary}
\end{aligned}
$$

where M and S are the coefficient matrices of the matching and side constraints, respectively.

A solution to (MS) yields a solution for (SP) given in

PROPOSITION 1: If $\{x_{jk}^*\}$ is an optimal solution of (MS) then $y_j^* = x_{j1}^*$, $j = 1, \ldots, n$ is an optimal solution to (SP).

Let $G(\lambda, x) = (c - \lambda S)x$ where the domain of x is $\{x \mid Mx = 1 \text{ and } x \text{ binary}\}$ and λ is an unrestricted vector of Lagrange multipliers. The Lagrangian relaxation of (MS) relative to $Sx = 0$ is

$$(LR_\lambda) \qquad F(\lambda) = \max_x G(\lambda, x).$$

Without matrix notation, the Lagrangian relaxation can be written as

$$\max \sum_{j=1}^{n} \sum_{k=1}^{K_j} (c_j - (\lambda_{jk} - \lambda_{j,k-1}))x_{jk}$$

$$\sum_{j=1}^{n} \sum_{k=1}^{K_j} a_{ij}^k x_{jk} = 1, \quad i = 1, \ldots, m$$

$$x_{jk} \in \{0, 1\}, \text{ all } j \text{ and } k$$

where, for $j = 1, \ldots, n$, λ_{j0} and λ_{jK_j} are defined to be zero. (LR_λ) is a (weighted, perfect) matching problem.

Relaxations play a very important role in integer programming algorithms. To be worthwhile, the relaxed problem should be easier to solve than the original one and should also yield a tight bound on the original problem solution. Lagrangian relaxations often fulfill both of these criteria. Since one of the criteria of a good relaxation is the tightness of its bound, the best choice for λ in (LR_λ) is the one that optimizes the Lagrangian dual

$$(LD) \qquad \min_\lambda F(\lambda)$$

where $\lambda = (\lambda_1, \ldots, \lambda_n)$ and $\lambda_j = (\lambda_{j1}, \ldots, \lambda_{j,K_j-1})$, for $j = 1, \ldots, n$.

Let $v(P)$ represent the optimal objective function value of any problem (P), and let $(SPLP)$ represent the linear programming relaxation of (SP). Proposition 2 (see Geoffrion [7]) summarizes the relationships between (SP), $(SPLP)$, (MS), (LR_λ) and (LD).

PROPOSITION 2:

(a) $v(SP) = v(\text{MS}) \leqslant v(LPSP)$,

(b) for all λ, $v(MS) \leqslant v(LR_\lambda)$ $(=F(\lambda))$,

(c) if for a given λ a vector x is optimal in (LR_λ) and $Sx = 0$, then x is an optimal solution of (MS), and

(d) $v(LD) \leqslant v(SPLP)$.

Note that the Lagrangian relaxation using the λ found in (LD) is at least as tight as the linear programming relaxation; this is a consequence of the fact that matrix M is not totally unimodular. Typically, $v(SP) < v(\text{LD}) < v(SPLP)$.

3. OPTIMIZING THE LAGRANGIAN DUAL

Many methods (surveyed in Fisher, Northup and Shapiro [6] and Bazaraa and Goode [2]) have been proposed for solving Lagrangian duals. By far the most widely used is the subgradient optimization method described in Held and Karp [8] and Held, Wolfe and Crowder [9]. Compared to other methods, very little "overhead" is needed and, most importantly, it has proven to be very effective computationally. In subgradient optimization, a sequence $\{\lambda^i\}$ of multiplier vectors is generated iteratively, using at each iteration the solution $F(\lambda^i)$. Many components of each λ^i change from iteration to iteration, and in the context of solving (LD), new optimal matchings must be solved "from scratch." Although solving a large matching problem is much easier than solving a large linear programming problem, it still can be time consuming.

The (weighted, perfect) matching problem

$$\max cx$$
$$Mx = 1$$
$$x \text{ binary}$$

where each column of M contains exactly two nonzero entries, both equal to one, can be interpreted graphically by letting M be the node-edge incidence matrix of a graph in which each row of M represents a node and each column represents an edge where edge k meets node i if and only if $m_{ik} = 1$, and c_k is the weight assigned to edge k. The problem then is to choose a set of edges, called a feasible matching, so that each node meets exactly one of the edges selected, in such a way that the sum of the weights on the edges chosen is as large as possible. Edmonds [3,4,5] developed an efficient (polynomially bounded) primal-dual algorithm for solving the matching problem and Weber [14] showed how sensitivity analysis can be performed on optimal matchings to get the new optimal solution from the original optimal solution if the weight on an edge is changed. Except for some very simple special cases, the techniques involve modifying the graph by attaching additional nodes and edges near the edge whose edge-weight is to be altered. Edmonds's algorithm is re-entered with all the needed properties, including complementary slackness, being maintained. The final primal and dual solutions for the modified problem are then "translated back" to yield the optimal matching for the single altered edge-weight problem, in such a way that again, all the needed properties are maintained (and thus, the process can be repeated if other edge-weights are altered). Reoptimizing using these techniques when a single edge-weight is altered is on the order of cardinality (N) more efficient than using Edmonds' algorithm "from scratch," where N equals the number of nodes in the graph.

Because of the special structure present in the side constraints of (MS) in which each variable appears in at most two equations, we choose to attempt to optimize (LD) by using an improved version of the cyclic coordinate method of nonlinear programming. The structure of the S matrix results in each λ_{jk} appearing in the coefficient of at most two variables in $G(\lambda, x)$. This allows the sensitivity analysis techniques to be used to improve significantly the usual cyclic coordinate method. In this method, $F(\lambda)$ is optimized cyclically in each of the coordinate directions. Thus, after initializing λ, we minimize $F(\lambda)$ with respect to $\lambda_{11}, \ldots, \lambda_{1,K_1-1}, \ldots, \lambda_{n1}, \ldots, \lambda_{n,K_n-1}$ in that order, one at a time. This process, which involves $\sum_{j=1}^{n} (K_j - 1)$ single variable minimizations, is repeated until the objective function stops decreasing. Typically, each one-variable minimization is accomplished by one of the iterative or grid type of procedures used in unconstrained optimization algorithms. However, because of the special structure of the problem, Theorem 1 provides a direct formula for each minimization, thus avoiding the time consuming "line searches." The proof of Theorem 1 appears in the Appendix.

THEOREM 1: Suppose $\dot{x}^*$ is the current optimal matching vector, λ^* is the current optimal Lagrange multiplier vector, $\hat{\lambda}$ and $\ddot{\lambda}$ are identical to λ^* except for the λ_{jk} component, and $\hat{\lambda}_{jk} = 1 + |c - \lambda^* S| \cdot 1$ and $\ddot{\lambda}_{jk} = -1 - |c - \lambda^* S| \cdot 1$, $\hat{x}$ maximizes $G(\hat{\lambda}, x)$ and $\ddot{x}$ maximizes $G(\ddot{\lambda}, x)$. An optimal λ^{**}_{jk} that minimizes $F(\lambda)$ with respect to λ_{jk} with all other components of λ fixed at their values in λ^* depends on x^*, $\hat{x}$ or $\ddot{x}$, and λ^* as follows:

CASE 1: If $x^*_{jk} = x^*_{j,k+1} = 1$ then $\lambda^{**}_{jk} = \lambda^*_{jk}$.

CASE 2: If $x^*_{jk} = x^*_{j,k+1} = 0$ then $\lambda^{**}_{jk} = \lambda^*_{jk}$.

CASE 3: If $x^*_{jk} = 1$ and $x^*_{j,k+1} = 0$ then

(a) if $\hat{x}_{jk} = 1$ and $\hat{x}_{j,k+1} = 0$ then
$\lambda^{**}_{jk} = \infty$ (LD is unbounded and MS is infeasible),

(b) if $\hat{x}_{jk} = 1$ and $\hat{x}_{j,k+1} = 1$ then
$\lambda^{**}_{jk} = \lambda^*_{jk} + [F(\lambda^*) - F(\hat{\lambda})]$,

(c) if $\hat{x}_{jk} = 0$ and $\hat{x}_{j,k+1} = 0$ then
$\lambda^{**}_{jk} = \lambda^*_{jk} + [F(\lambda^*) - F(\hat{\lambda})]$,

(d) if $\hat{x}_{jk} = 0$ and $\hat{x}_{j,k+1} = 1$ then
$\lambda^{**}_{jk} = \lambda^*_{jk} + (1/2)[(\hat{\lambda}_{jk} - \lambda^*_{jk}) - (F(\hat{\lambda}) - F(\lambda^*))]$.

CASE 4: If $x^*_{jk} = 0$ and $x^*_{j,k+1} = 1$ then

(a) if $\ddot{x}_{jk} = 0$ and $\ddot{x}_{j,k+1} = 1$ then
$\lambda^{**}_{jk} = -\infty$ (LD is unbounded and MS is infeasible),

(b) if $\ddot{x}_{jk} = 0$ and $\ddot{x}_{j,k+1} = 0$ then
$\lambda^{**}_{jk} = \lambda^*_{jk} - [F(\lambda^*) - F(\ddot{\lambda})]$,

(c) if $\ddot{x}_{jk} = 1$ and $\ddot{x}_{j,k+1} = 1$ then
$\lambda^{**}_{jk} = \lambda^*_{jk} - [F(\lambda^*) - F(\ddot{\lambda})]$,

(d) if $\ddot{x}_{jk} = 1$ and $\ddot{x}_{j,k+1} = 0$ then
$\lambda^{**}_{jk} = \lambda^*_{jk} - (1/2)[(\lambda^*_{jk} - \ddot{\lambda}_{jk}) - (F(\ddot{\lambda}) - F(\lambda^*))]$.

Theorem 1 is fairly easy to implement. The only unknown quantities in the formulas for λ^{**}_{jk} are $F(\hat{\lambda})$ and $F(\ddot{\lambda})$ and depending on x^*, at most one of these must be found. Computationally, the task of finding either one of these quantities is quite simple, since it is not necessary to solve a new matching problem "from scratch," but only to use sensitivity analysis techniques to reoptimize the matching with two edge-weights altered. The techniques are applied in those edges, one at a time. After computing λ^{**}_{jk}, the two edge-weights are again altered using λ^{**}_{jk} and a new optimal matching is determined using the sensitivity analysis techniques.

At each step of the cyclic coordinate method a new vector λ is generated, differing from the previous λ by at most one component. Let λ^i represent the i-th such vector.

THEOREM 2: Assuming v(MS) exists, the sequence $\{F(\lambda^i)\}$ converges.

PROOF: For all i, $F(\lambda^{i+1}) \leqslant F(\lambda^i)$ and, by Proposition 2, the sequence has a lower bound of $v(MS)$. A bounded, nonincreasing sequence has a limit. □

Let $\bar{F}$ denote the limit of $\{F(\lambda^i)\}$. Zangwill [15] and Luenberger [11] give mild restrictions, including $F(\lambda)$ having continuous first partial derivatives and a unique minimum point along any coordinate direction, that guarantee global convergence of the cyclic coordinate method. Unfortunately, $F(\lambda)$ violates these restrictions. It is not necessarily true that $\bar{F} = v(LD)$. In fact, if $F(\bar{\lambda}) = \bar{F}$ then $\lambda = \bar{\lambda}$ might not even be a local minimum, since it is only a relative minimum with respect to the coordinate directions.

The following is an example in which the sequence $\{F(\lambda^t)\}$ generated by the cyclic coordinate method does not converge to $v(LD)$.

Let O_5 be the null matrix of order 5 and

$$Z = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Let the A matrix of the set partitioning version of (SP) be the 20×25 matrix

$$A = \left[\begin{array}{ccccc} Z & O_5 & O_5 & O_5 & \begin{matrix}1&0&0&0&0\\1&0&0&0&0\\1&0&0&0&0\\1&0&0&0&0\end{matrix} \\ O_5 & Z & O_5 & O_5 & \begin{matrix}0&1&0&0&0\\0&1&0&0&0\\0&1&0&0&0\\0&1&0&0&0\end{matrix} \\ O_5 & O_5 & Z & O_5 & \begin{matrix}0&0&1&0&0\\0&0&1&0&0\\0&0&1&0&0\\0&0&1&0&0\end{matrix} \\ O_5 & O_5 & O_5 & Z & \begin{matrix}0&0&0&1&0\\0&0&0&1&0\\0&0&0&1&0\\0&0&0&1&0\end{matrix} \\ \begin{matrix}1&0&0&0&0\\0&1&0&0&0\\0&0&1&0&0\\0&0&0&1&0\end{matrix} & \begin{matrix}1&0&0&0&0\\0&1&0&0&0\\0&0&1&0&0\\0&0&0&1&0\end{matrix} & \begin{matrix}1&0&0&0&0\\0&1&0&0&0\\0&0&1&0&0\\0&0&0&1&0\end{matrix} & \begin{matrix}1&0&0&0&0\\0&1&0&0&0\\0&0&1&0&0\\0&0&0&1&0\end{matrix} & \begin{matrix}1&0&0&0&0\\1&0&0&0&0\\1&0&0&0&0\\1&0&0&0&0\end{matrix} \\ \begin{matrix}0&1&0&0&0\\0&0&1&0&0\\0&0&0&1&0\\0&0&0&0&1\end{matrix} & \begin{matrix}0&1&0&0&0\\0&0&1&0&0\\0&0&0&1&0\\0&0&0&0&1\end{matrix} & \begin{matrix}0&1&0&0&0\\0&0&1&0&0\\0&0&0&1&0\\0&0&0&0&1\end{matrix} & \begin{matrix}0&1&0&0&0\\0&0&1&0&0\\0&0&0&1&0\\0&0&0&0&1\end{matrix} & \begin{matrix}0&1&0&0&0\\1&0&0&0&0\\1&0&0&0&0\\1&0&0&0&0\end{matrix} \end{array}\right]$$

Let the objective function coefficients corresponding to the first 20 columns of A be 1 and for the last 5 columns be 0. The solutions are $v(SP) = 2$ where $y_1 = y_{20} = y_{22} = y_{23} = y_{24} = 1$, $y_j = 0$, otherwise, $v(SPLP) = 5$ where $y_j = 1/4$, $j = 1, \ldots, 20$ and $y_j = 0$, otherwise, $v(LD) = 2$ where $\lambda_{10,1} = -1/2$, $\lambda_{11,1} = 1/2$, $\lambda_{22,1} = 1$, $\lambda_{24,1} = -1$, $\lambda_{jk} = 0$, otherwise, $\{F(\lambda^t)\} \rightarrow \bar{F} = 3$ using the cyclic coordinate method (established empirically).

Thus, $v(SP) = v(LD) < \bar{F}(\lambda) < v(SPLP)$. Notice that the bound achieved from solving the Lagrangian dual, for which the subgradient method is successful (established empirically), and the bound from the cyclic coordinate method are both superior to the one obtained by using linear programming. It should be pointed out that in the usual implementation of subgradient optimization, global convergence is not guaranteed.

In addition to avoiding "from scratch" solutions to large matching problems, another important reason for choosing the cyclic coordinate method instead of the subgradient method is that subgradient optimization lacks an important property of the easy to perform cyclic coordinate method. In using subgradient optimization, the sequence $\{F(\lambda^t)\}$ is not monotone; it

an take quite a few iterations until any progress is made in minimizing $F(\lambda)$. Since in a branch-and-bound method we are more interested in getting a close approximation for $v(LD)$ n a short period of time than we are in solving it exactly, it seems reasonable to choose a nethod that begins showing progress in minimizing $F(\lambda)$ immediately. Actual computational omparisons of the two methods are given in the next section.

3. COMPUTATIONAL RESULTS

The results of the computational experiments performed only at the initial node of a branch-and-bound tree are summarized in three tables. Fourteen problems of varying sizes were run using the cyclic coordinate method (Table 1) and the subgradient method (Table 2) or optimizing the Lagrangian dual (LD), and using linear programming (Table 3) for solving he continuous relaxation of (SP). Each problem contains exactly four ones per column, and n the tables, each is labeled type S2, R1 or RR. S2 is the example given in Section 2. The ther two types have constraint matrices consisting of a randomly generated portion containing $n-m/4$ columns and a set of $m/4$ "dummy" columns to insure feasibility. The i-th such dummy" column contains ones in rows $4i-3$, $4i-2$, $4i-1$ and $4i$, and zeros elsewhere. The bjective function coefficients are zero for these columns. Types R1 and RR differ in the bjective function coefficients for the other columns. Problems of type R1 have all the oefficients equal to one, while problems of type RR have randomly generated integer oefficients with values between one and ten.

TABLE 1. *Cyclic Coordinate Method**

Problem	$m \times n$	Type of Data	Initial Value	Final Value	$Sx = 0$?	Cycle No. at Termination	Iterations	Time(sec.)† on IBM 370/168
1	20 × 20	R1	4.5	3	No	2	13	10
2	20 × 20	R1	5	3.98	No	4	30	15
3	20 × 25	S2	4	3‡	No	13	67	4.78
4	20 × 50	R1	5	5	No	5	40	15
5	20 × 50	R1	5	5	No	2	14	10
6	20 × 50	RR	40.5	35.00	No	5	32	30
7	40 × 40	R1	8.5	3.80	No	2	22	20
8	40 × 40	R1	9.5	8.25	No	1	16	20
9	40 × 100	R1	10	10	No	1	15	15
10	40 × 100	R1	10	10	No	1	17	20
11	40 × 100	RR	84.5	81.06	No	1	13	30
12	60 × 60	R1	13	9.70	No	2	29	60
13	60 × 150	R1	15	15	No	1	18	60
14	100 × 250	R1	25	25	No	1	5	60

*The program for the matching algorithm is given in [13].
†CPU time. Integer values indicate arbitrarily set CPU time limits.
‡Converging to within .00001 of 3.

TABLE 2. *Subgradient Method*

Problem	$m \times n$	Type of Data	Initial Value	Final Value	Best Value	$Sx = 0$?	Iterations	Time (sec.) on IBM 370/168
1	20 × 20	R1	4.5	0	0	Yes	9	1.45
2	20 × 20	R1	5	.60	.57	No	77	15
3	20 × 25	S2	4	2	2	Yes	9	1.27
4	20 × 50	R1	5	5.43	5	No	29	15
5	20 × 50	R1	5	5.58	5	No	34	15
6	20 × 50	RR	40.5	32.94	32.81	No	69	30
7	40 × 40	R1	8.5	0	0	Yes	10	5.98
8	40 × 40	R1	9.5	0	0	Yes	12	7.86
9	40 × 100	R1	10	18.33	10	No	13	20
10	40 × 100	R1	10	21.24	10	No	12	20
11	40 × 100	RR	84.5	105.06	84.5	No	17	30
12	60 × 60	R1	13	0	0	Yes	16	37.10
13	60 × 150	R1	15	34.15	15	No	11	60
14	100 × 250	R1	25	106.53	25	No	5	60

TABLE 3. *Linear Programming*†

Problem	$m \times n$	Type of Data	Final Value	Optimal?	Binary?	Iterations	Time (sec.) on IBM 370/168
1	20 × 20	R1	0	Yes	Yes	19	.29
2	20 × 20	R1	0	Yes	Yes	24	.34
3	20 × 25	S2	5	Yes	No	20	.31
4	20 × 50	R1	5	Yes	No	42	.58
5	20 × 50	R1	5	Yes	No	78	.80
6	20 × 50	RR	29.90	Yes	No	45	.73
7	40 × 40	R1	0	Yes	Yes	54	.65
8	40 × 40	R1	0	Yes	Yes	57	.74
9	40 × 100	R1	10	Yes	No	308	5.18
10	40 × 100	R1	10	Yes	No	350	6.18
11	40 × 100	RR	71.02	Yes	No	156	2.76
12	60 × 60	R1	0	Yes	Yes	93	1.23
13	60 × 150	R1	15	Yes	No	516	13.44
14	100 × 250	R1	<25.44	No	—	1209	60

†FORTRAN code given in Land and Powell [10].

In Table 1 a distinction is made between a cycle and an iteration. Each time $F(\lambda)$ is minimized with respect to all of $\lambda_{11}, \ldots, \lambda_{1,K_1-1}, \ldots, \lambda_{n1}, \ldots, \lambda_{n,K_n-1}$ in that order, one at a time while the others are fixed, a cycle is completed. However, if when minimizing $F(\lambda)$ with respect to say λ_{jk} we have that $x_{jk} \neq x_{j,k+1}$ then this is considered an iteration. Thus, there are potentially as many as $m/2$ iterations per cycle. Loosely speaking, a cycle in the cyclic coordinate method corresponds to an iteration in the subgradient method.

Very seldom does an algorithm perform uniformly better than another on all problems, and the three methods tested are no exception to this rule. Each out-performs a competing method on at least one of the fourteen problems tested. However, certain general observations can be made. The cyclic coordinate method performs much slower than anticipated, although it does do better than the subgradient method on problem 11. Not surprisingly, linear programming was highly successful in all randomly generated problems except problem 14, the largest one, in which it was inferior to the other two methods. For this problem, linear programming failed to reach an optimum in one minute, while the other two methods were each able to provide useful information since several matching problems were able to be solved in one minute.

We are not discouraged by the fact that the linear programming method out-performs the cyclic coordinate and subgradient methods on the majority of the test problems. The results of problem 3 indicate that there could be a class of problems in which, regardless of the size, the cyclic coordinate and subgradient methods are superior to linear programming, and perhaps more importantly, problem 14 indicates that perhaps for large problems the methods developed here could be a viable alternative to those algorithms that use linear programming.

ACKNOWLEDGMENT

We would like to thank Jack Edmonds for many helpful suggestions, particularly with regard to sensitivity analysis of the optimal matchings.

REFERENCES

[1] Balas, E. and M.W. Padberg, "Set Partitioning," pp. 205-258 in B. Roy, ed., *Combinatorial Programming: Methods and Applications*, (D. Reidel Publishing Co., 1975).

[2] Bazaraa, M.S. and J.J. Goode, "A Survey of Various Tactics for Generating Lagrangian Multipliers in the Context of Lagrangian Duality," School of Industrial and Systems Engineering, Georgia Institute of Technology (1974).

[3] Edmonds, J., "Path, Trees, and Flowers," Canadian Journal of Mathematics *17*, 449-467 (1965).

[4] Edmonds, J., "Maximum Matching and a Polyhedron with 0,1-Vertices," Journal of Research of the National Bureau of Standards *69B*, 125-130 (1965).

[5] Edmonds, J., "An Introduction to Matching," notes on lectures given at Ann Arbor, Michigan (1967).

[6] Fisher, M.L., W.D. Northup and J.F. Shapiro, "Using Duality to Solve Discrete Optimization Problems: Theory and Computational Experience," Mathematical Programming Study *3*, 56-94 (1975).

[7] Geoffrion, A.M., "Lagrangian Relaxation for Integer Programming," Mathematical Programming Study *2*, 82-114 (1974).

[8] Held, M. and R.M. Karp, "The Traveling-Salesman Problem and Minimum Spanning Trees: Part II," Mathematical Programming *1*, 6-25 (1971).

[9] Held, M., P. Wolfe and H.P. Crowder, "Validation of Subgradient Optimization," Mathematical Programming *6*, 62-88 (1974).

[10] Land, A.H. and S. Powell, *Fortran Codes for Mathematical Programming: Linear, Quadratic and Discrete* (John Wiley and Sons, 1973).
[11] Luenberger, D.G., *Introduction to Linear and Nonlinear Programming*, (Addison-Wesley, 1973).
[12] Marsten, R.E., "An Algorithm for Large Set Partitioning Problems," Management Science *20*, 774-787 (1974).
[13] Weber, G.M., "A Solution Technique for Binary Integer Programming Using Matchings on Graphs," Ph.D. Thesis, Cornell University (1978).
[14] Weber, G.M., "Sensitivity Analysis of Optimal Matchings," TR No. 427, School of Operations Research and Industrial Engineering, Cornell University (May 1979).
[15] Zangwill, W.I., *Nonlinear Programming: A Unified Approach* (Prentice-Hall, 1969).

APPENDIX

PROOF OF THEOREM 1: The proof of Case 2 parallels Case 1 and Case 4 parallels Case 3; thus the proofs of only Cases 1 and 3 are given.

Throughout the proof we use the fact that a change in λ^*_{jk} to $\tilde{\lambda}_{jk}$ changes the objective function coefficients of x_{jk} and $x_{j,k+1}$ in (LR_λ) by $\lambda^*_{jk} - \tilde{\lambda}_{jk}$ and $\tilde{\lambda}_{jk} - \lambda^*_{jk}$, respectively.

CASE 1: By definition $F(\lambda^{**}) = \max_x G(\lambda^{**}, x) \geqslant G(\lambda^{**}, x^*)$. Since $x^*_{jk} = x^*_{j,k+1} = 1$, for any λ identical to λ^* except for the λ_{jk} component, $G(\lambda, x^*) = G(\lambda^*, x^*) = F(\lambda^*)$. In particular, $G(\lambda^{**}, x^*) = F(\lambda^*)$ so that $F(\lambda^{**}) \geqslant F(\lambda^*)$. Now $F(\lambda^{**}) = \min_\lambda \{F(\lambda)$: $\lambda = \lambda^*$ except for the λ_{jk} component$\} \leqslant F(\lambda^*)$. Hence $F(\lambda^{**}) = F(\lambda^*)$ and $\lambda^{**} = \lambda^*$.

CASE 3: Let $\lambda = \lambda^*$ and consider continuously increasing λ_{jk} from λ^*_{jk} and altering the matching x^* only if $G(\lambda, x)$ would increase by doing so. Let $\bar{\lambda}_{jk}$ be the value of λ_{jk} when x_{jk} first becomes 0, if such a λ_{jk} exists, otherwise set $\bar{\lambda}_{jk} = \infty$. Let $\bar{\bar{\lambda}}_{jk}$ be the value of λ_{jk} when $x_{j,k+1}$ first becomes 1, if such a λ_{jk} exists, otherwise set $\bar{\bar{\lambda}}_{jk} = \infty$. Let $a = \min(\bar{\lambda}_{jk}, \bar{\bar{\lambda}}_{jk})$ and $b = \max(\bar{\lambda}_{jk}, \bar{\bar{\lambda}}_{jk})$. Note that $\hat{\lambda}_{jk}$ has been chosen sufficiently large so that if a or b are finite, they are smaller than $\hat{\lambda}_{jk}$. Thus

$$\hat{x}_{jk} = \begin{cases} 0 & \bar{\lambda}_{jk} \text{ finite} \\ 1 & \text{otherwise} \end{cases} \qquad \hat{x}_{j,k+1} = \begin{cases} 0 & \text{otherwise} \\ 1 & \bar{\bar{\lambda}}_{jk} \text{ finite.} \end{cases}$$

As long as $x_{jk} = 1$, there is a unit decrease in the objective function per unit increase in λ_{jk} and when $x_{j,k+1} = 1$ there is a unit increase in the objective function per unit increase in λ_{jk}. Thus,

$$\partial F(\lambda)/\partial \lambda_{jk} = \begin{cases} -1, & \text{if } \lambda^*_{jk} \leqslant \lambda_{jk} < a \\ 0, & \text{if } a < \lambda_{jk} < b \\ 1, & \text{if } b < \lambda_{jk} \leqslant \hat{\lambda}_{jk}. \end{cases} \tag{1}$$

In Case 3(a), we have $a = \infty$ so that $F(\lambda)$ decreases monotonically with λ_{jk} and the dual is unbounded ($\lambda^{**}_{jk} = \infty$).

In Cases 3(b,c), a is finite and $b = \infty$. We can set $\lambda_{jk}^{**} = a$. From (1), $F(\hat{\lambda}) - F(\lambda^*) = -(a - \lambda_{jk}^*)$ so that

$$\lambda_{jk}^{**} = a = \lambda_{jk}^* + F(\lambda^*) - F(\hat{\lambda}).$$

In Case 3(d), a and b are finite. We can set $\lambda^{**} = (a + b)/2$. From (1), $F(\hat{\lambda}) - F(\lambda^*) = -(a - \lambda_{jk}^*) + (\hat{\lambda}_{jk} - b)$ so that

$$\lambda_{jk}^{**} = (a + b)/2 = (\lambda_{jk}^* + \hat{\lambda}_{jk} + F(\lambda^*) - F(\hat{\lambda}))/2. \qquad \square$$

A COMPLETE IMPORTANCE RANKING FOR COMPONENTS OF BINARY COHERENT SYSTEMS, WITH EXTENSIONS TO MULTI-STATE SYSTEMS

David A. Butler

Oregon State University
Corvallis, Oregon

ABSTRACT

Means of measuring and ranking a system's components relative to their importance to the system reliability have been developed by a number of authors. This paper investigates a new ranking that is based upon minimal cuts and compares it with existing definitions. The new ranking is shown to be easily calculated from readily obtainable information and to be most useful for systems composed of highly reliable components. The paper also discusses extensions of importance measures and rankings to systems in which both the system and its components may be in any of a finite number of states. Many of the results about importance measures and rankings for binary systems are shown to extend to the more sophisticated multi-state systems. Also, the multi-state importance measures and rankings are shown to be decomposable into a number of sub-measures and rankings.

Given a system composed of many components, a question of considerable interest is which components are most crucial to the proper functioning of the system. In response to this question, a number of importance measures and rankings have been proposed [3], [4], [5], [10]. This paper investigates a new ranking and compares it to existing rankings, principally the ranking induced by the Birnbaum reliability importance measure. The new ranking is based upon minimal cuts and provides a complete ordering of all the system's components relative to their importance to the system reliability. This ranking has three key points in its favor: (i) the calculations involved require only readily obtainable information; (ii) the calculations are usually quite simple; and (iii) the ranking is designed for use with systems consisting of highly reliable components, the most common case.

The final section of the paper deals with extensions of importance measures and rankings to systems in which both the system and its components may be in any of a finite number of states. Many of the results about importance measures and rankings for binary systems established in preceding sections are shown to extend to the more sophisticated multi-state systems. Also, the multi-state importance measures and rankings are shown to be decomposable into a number of sub-measures and rankings.

1. DEFINITIONS OF COMPONENT IMPORTANCE—BINARY SYSTEMS

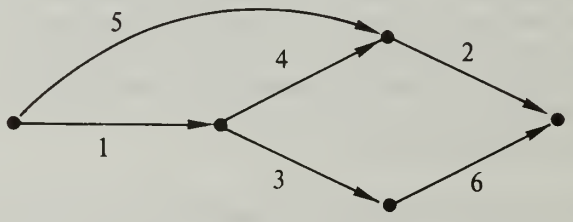
Consider a binary coherent system of n independent components with structure function $\phi(\underline{x})$ and reliability function $h(\underline{p})$. (Our definitions, terminology and notation regarding binary coherent systems follow those of [2].)

Birnbaum [4] and Barlow and Proschan [3] have each proposed reliability importance measures, so called because they make use of probabilistic information about the components. The Birnbaum reliability measure of the importance of component i is $I_h(i;\underline{p}) = \partial h(\underline{p})/\partial p_i$. The Barlow-Proschan reliability measure requires the time-to-failure distribution for each component, and the importance of a component i using this measure can be interpreted as the probability that component i causes the system to fail [3].

The two sets of authors have also proposed structural importance measures, so called because they require only a knowledge of the system structure function to be calculated. This feature gives them an important practical advantage over the more sophisticated reliability importance measures, because often the more detailed knowledge required for the calculation of these latter measures is unobtainable. Both structural measures can be derived from the Birnbaum reliability importance measure, assuming a common reliability p for all components. Specifically, the Birnbaum structural importance measure is the Birnbaum reliability importance measure evaluated at $p = 0.5$ [4]. The Barlow-Proschan structural importance measure [3] is the "average" (integral) of the Birnbaum reliability measure as p ranges over [0,1]. But for most systems the typical component's reliability is not 0.5 or even 0.5 "on the average," but rather is much higher. This is especially so for systems with complex structure functions incorporating redundancy, because redundancy in the design of a system is usually only incorporated if a non-redundant design with highly reliable components cannot produce a satisfactory level of system reliability. Thus using either of these two measures to compare components of such systems may give a misleading picture of which components are most important. It would seem desirable, therefore, to develop a measure or ranking that is structural (i.e., is based solely upon the system structure function and therefore not upon $\underline{p}$), yet is somehow related to the Birnbaum reliability importance measure for high values of $\underline{p}$. The ranking proposed in this paper is such a result. This ranking is based upon cuts and provides a complete ordering of all components. It extends an earlier ranking which provided only a partial ordering of the components [5].

To introduce this ranking, consider the following example.

EXAMPLE 1.



$$\phi(\underline{x}) = [1 - (1 - x_1)(1 - x_5)] \cdot [1 - (1 - x_2)(1 - x_3)] \cdot$$
$$[1 - (1 - x_2)(1 - x_6)] \cdot [1 - (1 - x_4)(1 - x_5)(1 - x_6)] \cdot$$
$$[1 - (1 - x_3)(1 - x_4)(1 - x_5)] \cdot [1 - (1 - x_1)(1 - x_2)]$$

Min cuts: $C_1 = \{1,5\}$, $C_2 = \{2,3\}$, $C_3 = \{2,6\}$
$C_4 = \{4,5,6\}$, $C_5 = \{3,4,5\}$, $C_6 = \{1,2\}$.

Assuming that the components function or fail independently of one another and assuming a common reliability p for each component, the Birnbaum reliability importances can be written as

$$I_h(1;\ p) = 2(1-p) - 3(1-p)^2 - (1-p)^3 + 3(1-p)^4 - (1-p)^5$$

$$I_h(2;\ p) = 3(1-p) - 4(1-p)^2 - (1-p)^3 + 3(1-p)^4 - (1-p)^5$$

$$I_h(3;\ p) = (1-p) - (1-p)^2 - 2(1-p)^3 + 3(1-p)^4 - (1-p)^5$$

$$I_h(4;\ p) = 2(1-p)^2 - 5(1-p)^3 + 4(1-p)^4 - (1-p)^5$$

$$I_h(5;\ p) = (1-p) + (1-p)^2 - 5(1-p)^3 + 4(1-p)^4 - (1-p)^5$$

$$I_h(6;\ p) = (1-p) - (1-p)^2 - 2(1-p)^2 + 3(1-p)^4 - (1-p)^5$$

Denote the component orderings induced by the Birnbaum structural importance measure, the Barlow-Proschan structural importance measure, and the Birnbaum reliability importance measure by $>_{BS}$, $>_{BPS}$, $>_{BR}$, respectively. (Note that $>_{BR}$ depends implicitly upon p.) Then

$$2 >_{BS} 5 >_{BS} 1 >_{BS} 3 =_{BS} 6 >_{BS} 4$$

$$2 >_{BPS} 5 >_{BPS} 1 >_{BPS} 3 =_{BPS} 6 >_{BPS} 4$$

and

$$2 >_{BR} 5,\ 1 >_{BR} 3 =_{BR} 6 >_{BR} 4 \text{ for all } \underline{p} = (p,p,\ldots,p).$$

For $p < (-1+\sqrt{5})/2 \approx .618$, $5 >_{BR} 1$ and the three rankings are indentical. But for $p > (-1+\sqrt{5})/2$, $1 >_{BR} 5$.

Notice that if the Birnbaum reliability measures $I_h(i;\ p)$ are written as polynomials in $(1-p)$ as above, then for high values of p the lowest-order terms in the polynomial dominate the rest. Thus, in this example, by looking only at the lowest-order terms in the formulas for $I_h(i;\ p)$ it is apparent that for high values of p,

$$2 >_{BR} 1, \quad 1 >_{BR} 5, \quad 1 >_{BR} 3, \quad 1 >_{BR} 6,$$
$$5 >_{BR} 4, \quad 3 >_{BR} 4, \quad 6 >_{BR} 4.$$

By examining the lowest- and the second lowest-order terms, we can further determine that for high values of p

$$5 >_{BR} 3 \text{ and } 5 >_{BR} 6.$$

This suggests a possible way to define a new structural ranking that would agree with $>_{BR}$ for high values of p. It will be more convenient to define this new structural ranking in terms of the system's min cuts, rather than the coefficients in the polynomial expressions for $I_h(i;\ p)$. However, we will see that the resulting definition is equivalent to the above.

DEFINITION: For each component l of a coherent system (N, ϕ) with t minimal cuts, let $d_{ij}^{(l)}$ denote the number of collections of i distinct min cuts such that the union of each collection contains exactly j components and includes component l, $(1 \leqslant i \leqslant t,\ 1 \leqslant j \leqslant n)$. Let $b_j^{(l)} = \sum_{i=1}^{t} (-1)^{i-1} d_{ij}^{(l)}$. Let $\underline{b}^{(l)} = (b_1^{(l)}, \ldots, b_n^{(l)})$. Component l is *more cut-important* than component k, denoted $l >_c k$, if and only if $\underline{b}^{(l)} \succ \underline{b}^{(k)}$, where $\succ$ denotes lexicographic ordering. Components l and k are *equally cut-important*, denoted $l =_c k$, if and only if $\underline{b}^{(l)} = \underline{b}^{(k)}$.

This definition, although rather formidable in appearance, is in practice usually easy to apply and work with. Because the ranking depends upon a lexicographic ordering, most components can be ranked by determining only the first few components of $\underline{b}^{(\cdot)}$. Also, the first non-zero component of any $\underline{b}^{(\cdot)}$ is particularly easy to compute (see Proposition 1 and Corollary 1).

EXAMPLE 1. (continued)

For $l = 2$, the non-zero $d_{ij}^{(l)}$'s are as follows: $d_{12}^{(2)} = 3$, $d_{23}^{(2)} = 4$, $d_{24}^{(2)} = 4$, $d_{25}^{(2)} = 4$, $d_{34}^{(2)} = 3$, $d_{35}^{(2)} = 11$, $d_{36}^{(2)} = 5$, $d_{45}^{(2)} = 4$, $d_{46}^{(2)} = 11$, $d_{56}^{(2)} = 6$, $d_{66}^{(2)} = 1$. Thus $\underline{b}^{(2)} = (0,3,-4,-1,3,-1)$. Similarly,

$$\underline{b}^{(1)} = (0,2,-3,-1,3,-1), \quad \underline{b}^{(3)} = (0,1,-1,-2,3,-1),$$
$$\underline{b}^{(4)} = (0,0,2,-5,4,-1), \quad \underline{b}^{(5)} = (0,1,1,-5,4,-1),$$
$$\underline{b}^{(6)} = (0,1,-1,-2,3,-1).$$

Therefore $2 >_c 1 >_c 5 >_c 3 =_c 6 >_c 4$.

Note that $\underline{b}^{(j)}$ is the vector of coefficients in the polynomial expression for $I_h(j; p)$. We will show that this is so in general. Also note that the cut-importance ranking and the Birnbaum reliability importance ranking for high values of p are in agreement.

2. ANALYSIS OF THE CUT-IMPORTANCE RANKING—BINARY SYSTEMS

As stated in the introduction, the cut-importance ranking has three main favorable properties: (i) it is based upon readily obtainable information, (ii) is usually easily calculated, and (iii) is designed for use when component reliabilities are high. The first property is already established, since this ordering is based only upon the system structure function through the minimal cuts of the system. This section will deal with the second and third properties.

The precise meaning of the third property of the cut-importance ranking is given in Theorems 1 and 2 below. The first theorem relates the cut-importance ranking to the Birnbaum reliability importance measure in the case where the component reliabilities are equal and high.

THEOREM 1: For $\underline{p} = (p,p, \ldots, p)$ where the scalar p is sufficiently close to one, the orderings $>_a$ and $>_c$ are identical.

PROOF: The above is a direct result of Lemma 1 which follows. Using the lemma, it is clear that $l =_c k$ if and only if $I_h(l; p) = I_h(k;p)$ for all p. [A scalar second argument is used in $I_h(k; p)$ when $\underline{p} = (p, \ldots, p)$.] Also, $l >_c k$ if and only if $\underline{b}^{(l)} - \underline{b}^{(k)} \succ \underline{0}$, and $\underline{b}^{(l)} - \underline{b}^{(k)} \succ \underline{0}$ if and only if $I_h(l; p) > I_h(k; p)$ for all p sufficiently close to one. □

LEMMA 1:
$$I_h(l; p) = \sum_{j=1}^{n} b_j^{(l)} (l - p)^{j-1}.$$

PROOF:
$$h(\underline{p}) = Pr\left(\bigcap_{i=1}^{t} E_i \right),$$

where E_i denotes the event that at least one component in i^{th} min cut functions. Thus

$$h(\underline{p}) = 1 - Pr\left(\bigcup_{i=1}^{t} E_i^c \right).$$

By the inclusion-exclusion principle ([8]; pp 98-101),

$$h(\underline{p}) = 1 - \sum_{i=1}^{t} (-1)^{i-1} S_i,$$

where

$$S_i = \sum_{1 \leqslant j_1 < j_2 < \cdots < j_i \leqslant t} Pr\left(E_{j_1}^c \cap E_{j_2}^c \cap \cdots \cap E_{j_i}^c\right).$$

Now the event $E_{j_1}^c \cap E_{j_2}^c \cap \cdots \cap E_{j_i}^c$ is the event that all components $k \in C_{j_1} \cup C_{j_2} \cup \cdots \cup C_{j_i}$ fail.

Thus using the independence assumption,

$$S_i = \sum_{1 \leqslant j_1 < j_2 < \cdots < j_i \leqslant t} \left[\prod_{k \in C_{j_1} \cup C_{j_2} \cup \cdots \cup C_{j_i}} (1 - p_k) \right]$$

where $C_1, \ldots, C_t$ are the minimal cuts of the system. Thus

$$I_h(l;\underline{p}) = \frac{\partial h(\underline{p})}{\partial p_l} = \sum_{i=1}^{t} (-1)^{i-1} \sum_{\substack{1 \leqslant j_1 < j_2 < \cdots < j_i \leqslant t \\ l \in C_{j_1} \cup C_{j_2} \cup \cdots \cup C_{j_i}}} \left[\prod_{\substack{k \in C_{j_1} \cup C_{j_2} \cup \cdots \cup C_{j_i} \\ k \neq l}} (1 - p_k) \right].$$

Recalling $p_1 = p_2 = \ldots = p_n = p$, and the definition of $d_{ij}^{(l)}$,

$$I_h(l;p) = \sum_{i=1}^{t} \sum_{j=1}^{n} (-1)^{i-1} (1-p)^{j-1} d_{ij}^{(l)},$$

$$= \sum_{j=1}^{n} b_j^{(l)} (1-p)^{j-1}.$$

□

Theorem 1 establishes the relationship the cut-importance ranking has to the Birnbaum reliability importance ranking in the case of high and equal component reliabilities. We now consider the case where the component reliabilities are high but unequal. Let $\underline{p}(\epsilon)$ be a vector-valued function of the positive scalar ϵ for which $0 < p_i(\epsilon) < 1$ for all $\epsilon \in (0, \infty)$ and $1 \leqslant i \leqslant n$. Let $\lim_{\epsilon \to 0} \underline{p}(\epsilon) = \underline{1}$. Unfortunately, it is not true in general that the component ordering induced by $I_h(\cdot;\underline{p}(\epsilon))$ coincides with $>_c$ for all ϵ sufficiently close to zero. However, with some additional assumptions on $\underline{p}(\epsilon)$ some partial results along these lines are possible. First we establish a simple and computationally convenient formula for the first non-zero coordinate in any vector $\underline{b}^{(l)}$.

PROPOSITION 1: For each component k, let e_k be the cardinality of the smallest minimal cut containing component k, and let f_k be the number of minimal cuts of cardinality e_k containing k. Then (i) $e_k = \min\{j : b_j^{(k)} \neq 0\}$, and (ii) $f_k = b_{e_k}^{(k)}$.

PROOF: By definition $d_{le_k}^{(k)} = f_k$. Also any union of two or more minimal cuts at least one of which contains k must have cardinality at least $e_k + 1$. Thus $d_{ie_k}^{(k)} = 0$ for all $i \geqslant 2$. Therefore

$$b_{e_k}^{(k)} = \sum_{i=1}^{t} (-1)^{i-1} d_{ie_k}^{(k)} = f_k.$$

Also, since component k is contained in no cuts of cardinality smaller than e_k, $d_{ij}^{(k)} = 0$ for all $j < e_k$. Thus $b_j^{(k)} = 0$ for $j < e_k$.

□

COROLLARY 1: (i) If $e_l < e_k$, then $l >_c k$.

(ii) If $e_l = e_k$ and $f_l > f_k$, then $l >_c k$.

THEOREM 2: Assume that for some M_1, $M_2 \in \mathbb{R}$,

$$\frac{1}{M_1} \leqslant \frac{q_1(\epsilon)}{q_j(\epsilon)} \leqslant \frac{1}{M_2} \text{ for all sufficiently small } \epsilon.$$

If either (i) $e_l < e_k$, or (ii) $e_l = e_k$ and $(f_l/f_k) > (M_1/M_2)^{e_k - 1}$, then there exists an $\hat{\epsilon} > 0$ such that $I_h(l;\underline{p}(\epsilon)) > I_h(k;\underline{p}(\epsilon))$ for all $\epsilon < \hat{\epsilon}$.

PROOF: See Theorem 2 in [5]. Further results along these lines are surely possible, but their value is questionable because the hypotheses become too complex. From a practical standpoint, users of the cut-importance ranking should be aware that while the cut-importance ranking can be useful even when component reliabilities are unequal, it may be misleading if the differences in the orders of magnitude of the unreliabilities are too great.

To summarize, the reliability importance measures and rankings probably give generally superior results to the structural ones and should be used unless (i) the probabilistic information required for their calculation is not available or (ii) computations involved are prohibitively extensive. However when one or the other of these conditions prevails, a structural measure or ranking must be employed. The Birnbaum or the Barlow-Proschan structural measure can be used but we have seen that doing so is equivalent to using the Birnbaum reliability importance measure $I_h(i;p)$ with $p = 0.5$ or 0.5 "on the average." If one feels that the component reliabilities, although not precisely known, are high, then the cut-importance ranking seems preferable, since, as Theorems 1 and 2 have shown, its results are the same as those given by the Birnbaum reliability importance ranking for high values of $\underline{p}$.

We now turn to the question of the computational complexities involved in determining the cut-importance ranking of a system's components. It is clear that the task of computing the entire vector $\underline{b}^{(k)}$ for each component k can be a formidable one for a complex system with many minimal cuts. However, Proposition 1 and Corollary 1 show that components often can be compared by only determining the easily computed quantities e_k and f_k. For instance, in Example 1 it is possible to determine that

$$2 >_c 1 >_c 5,\ 3,\ 6 >_c 4$$

in this manner. Also, since the structure function is symmetric in x_3 and x_6, it is clear that $3 =_c 6$. Thus additional calculations are necessary only to compare components 3 and 5. The ordering of these two components can be determined by computing the next entries in $\underline{b}^{(3)}$ and $\underline{b}^{(5)}$, namely $b_3^{(3)}$ and $b_3^{(5)}$. The last three entries in each vector $\underline{b}^{(\cdot)}$ are irrelevant for the purposes of ranking the components in this example.

In general, most components can be compared by determining the first non-zero entry in $\underline{b}^{(\cdot)}$ via Corollary 1. Other entries in $\underline{b}^{(\cdot)}$ are computed only as necessary.

Computations can also be simplified when the system under consideration contains modules.

PROPOSITION 2: Let (A, χ) be a module of (N, ϕ) and let $\phi(\underline{x}) = \psi(\chi(\underline{x}^A), \underline{x}^{A^c})$. Let ${}^{\phi}\underline{b}^{(\cdot)}$, ${}^{\chi}\underline{b}^{(\cdot)}$, and ${}^{\psi}\underline{b}^{(\cdot)}$ be the $\underline{b}^{(\cdot)}$ vectors corresponding to the structures ϕ, χ, and ψ, respectively. Then

$$^{\phi}\underline{b}_j^{(k)} = \sum_{i=1}^{j} {}^{\psi}\underline{b}_i^{(1)} \cdot {}^{x}\underline{b}_{j-i+1}^{(k)} \qquad \text{for all } k \in A,$$

where the definition of ${}^{x}\underline{b}^{(\cdot)}$ is extended to include zero coordinates for $i > |A|$, and ${}^{\psi}\underline{b}^{(1)}$ is extended similarly. (The above equation is just an expression of the fact that ${}^{\phi}\underline{b}^{(k)}$ is the convolution of the finite sequences ${}^{\psi}\underline{b}^{(1)}$ and ${}^{x}\underline{b}^{(k)}$.)

PROOF: In the remainder of the paper the dependence of $I_h(i;p)$ upon p will at times be suppressed and the notation simplified to $I_h(i)$. Let $I_h^{\phi}(\cdot)$, $I_h^{x}(\cdot)$, and $I_h^{\psi}(\cdot)$ denote the respective Birnbaum reliability importance measures. These three quantities are related as follows [4].

$$I_h^{\phi}(k) = I_h^{\psi}(1) \cdot I_h^{x}(k) \qquad \text{for all } k \in A.$$

Thus by Lemma 1,

$$\sum_{j=1}^{n} {}^{\phi}b_j^{(k)}(1-p)^{j-1} = \left[\sum_{j=1}^{|A|+1} {}^{\psi}b_j^{(1)}(1-p)^{j-1}\right]\left[\sum_{j=1}^{|A|} {}^{x}b_j^{(k)}(1-p)^{j-1}\right]$$

$$= \sum_{j=1}^{n}\left[\sum_{i=1}^{j} {}^{\psi}b_i^{(1)} \cdot {}^{x}b_{j-i+1}^{(k)}\right](1-p)^{j-1}.$$

Since this equality holds for all $0 \leqslant p \leqslant 1$, each pair of coefficients of the two polynomials must be identical.

□

Proposition 2 can be applied to make the calculation of the cut-importance component ranking simpler when the system contains modules.

EXAMPLE 1. (continued) Components 3 and 6 form a module.

$$A = \{3,6\},\ \chi(\underline{x}^{A}) = x_3x_6,\ \psi(z,\underline{x}^{A^c}) = 1 - (1 - x_1x_2x_4)(1 - x_1z)(1 - x_2x_5).$$

$$^{x}\underline{b}^{(3)} = (1,-1),\ ^{x}\underline{b}^{(6)} = (1,-1),\ ^{\psi}\underline{b}^{(1)} = (0,1,0,-2,1).$$

By using the concept of the dual of a coherent system, it is possible to develop a component ranking analogous to $>_c$, but based upon minimal paths instead of minimal cuts. This ordering can be shown to be identical to the ordering induced by $I_h(\cdot;p)$ when p is sufficiently small.

5. COMPONENT IMPORTANCE IN MULTI-STATE SYSTEMS

This section deals with extensions of the results of the preceding sections to systems in which both the system and its components may be in any of a finite number of states. Of course, any such increase in the sophistication of the model used to represent a real system entails disadvantages as well as advantages, and this extension to multi-state models is not intended to suggest that binary models are generally inadequate. To the contrary, in most cases they suffice quite well. However, in some instances a small increase in the number of states (say, to three or perhaps four) can result in a much improved model.

One of the main difficulties with multi-state models is the increased notational complexity. For this reason and for the reason that the number of states in a practical model must be kept small if the model is to be manageable, the following definitions will be given for ternary (three-state) systems; however, they will be given in a manner that illustrates the extension to

general n-state systems. Whenever the extension of a definition or result to n-state systems is unclear, some further explanation will be given.

The study of multi-state systems is a relatively new area in reliability theory. Most articles in this area have dealt with generalizing particular classes of results [1],[7],[9],[11],[12]. The most general paper in the area is Barlow's [1]. Let X_j denote the state of component j, $(X_j = 0, 1, 2,\ 1 \leqslant j \leqslant n)$. Given a collection of minimal cuts $C_1, C_2, \ldots, C_l$ which define the system structure, Barlow defines the system state $\phi(\underline{X})$ as the state of the "best" component in the "worst" min cut, i.e.,

$$\phi(\underline{X}) = \min_{1 \leqslant i \leqslant l} \{\max_{j \in C_i} \{X_j\}\}.$$

Let $Z_j = I_{\{X_j \geqslant k\}}$ and let $\psi = I_{\{\phi(\underline{X}) \geqslant k\}}$. Both $\underline{Z}$ and ψ are binary, and ψ is a function only of $\underline{Z}$. Because of this property, most results about binary coherent systems have immediate generalization under Barlow's extended definition. However, there are many more reasonable choices for the structure function ϕ in a multi-state setting than Barlow's definition allows (see [6] for some examples). To accommodate such choices, a more general definition of a multi-state coherent system is proposed below.

Let $S = \{\underline{x} \in \mathbb{R}^n : x_i = 0, 1, 2\}$, and let $(\cdot_i, \underline{x}) = (x_1, x_2, \ldots, x_{i-1}, \cdot, x_{i+1}, \ldots, x_n)$.

DEFINITION: Component i is relevant if and only if $\phi(2_i, \underline{x}) \neq \phi(0_i, \underline{x})$ for some $\underline{x} \in S$. Otherwise component i is *irrelevant*. Component i is *fully relevant* if and only if $\phi(2_i, \underline{x}) \neq \phi(1_i, \underline{x})$ for some $\underline{x} \in S$ and $\phi(1_i, \underline{y}) \neq \phi(0_i, \underline{y})$ for some $\underline{y} \in S$.

DEFINITION: A structure function ϕ is *coherent* if and only if

(i) $\phi(\underline{0}) = 0$; $\phi(\underline{2}) = 2$,

(ii) $\phi(\underline{x})$ is non-decreasing in $\underline{x}$,

(iii) each component is relevant.

The ordered pair (N, ϕ) is called a *(generalized or ternary) coherent system.*

If a component is not fully relevant, then only two states are required to describe its status. Such components are permissible in a generalized coherent system to allow for a mixture of binary and ternary components.

Define the matrix $P = [p_{ij}]$ by

$$P_{ij} = Pr\{\text{component } i \text{ is in state } j\}, \quad 1 \leqslant i \leqslant n,\ 0 \leqslant j \leqslant 2.$$

The reliability function, $h(P)$, is defined by

$$h(P) = Pr\{\phi(\underline{X}) \geqslant m\},$$

where $0 < m \leqslant 2$. The effect of this generalized definition of the reliability function is to consider systems whose components have several states but whose structure function effectively has two states ($\geqslant m$ or $< m$). All subsequent definitions and results are for a fixed value of m. (For simplicity, the dependence of $h(\cdot)$ upon m is suppressed in the notation.) For any matrix $A = [a_{ij}]$, let (k_i, A) denote the matrix whose $i-j$th entry is given by

$$(k_l, A)_{ij} = \begin{cases} a_{ij} & i \neq l, \\ 1 & i = l,\ j = k \\ 0 & i = l,\ j \neq k. \end{cases}$$

DEFINITION: The *r,s reliability importance of component i,* denoted by $I_h^{r,s}(i;P)$, is given by

$$I_h^{r,s}(i;\ P) = h(r_i, P) - h(s_i, P),$$

where $r,s = 0,1,2$ and $r > s$. The 2,0 reliability importance will sometimes be simply called the *reliability importance* and be denoted by $I_h(i;\ P)$. The *r,s* reliability importance of component *i* is the probability that the system is in state *m* or better given component *i* is in state *r* minus the probability that the system is in state *m* or better given component *i* is in state *s*.

DEFINITION: A vector $\underline{x} \in S$ is *r,s critical for component i* if and only if $\phi(r_i, \underline{x}) \geqslant m$ and $\phi(s_i, \underline{x}) < m$. $(r,s = 0,1,2;\ r > s)$

DEFINITION: Let $n_\phi^{r,s} = |\{\underline{x} \in S: \underline{x} \text{ is } r,s \text{ critical for component } i\}|$. The *r,s structural importance of component i,* $I_\phi^{r,s}(i)$, is given by

$$I_\phi^{r,s}(i) = 3^{-n}\, n_\phi^{r,s}(i).$$

In the following, whenever a vector $\underline{p} \in \mathbb{R}^3$ appears in an expression normally involving the matrix P, P will be understood to be the matrix all of whose rows are equal to $\underline{p}$.

PROPOSITION 3:
(i) $I_h^{2,0}(i) = I_h^{2,1}(i) + I_h^{1,0}(i)$
(ii) $I_\phi^{2,0}(i) = I_\phi^{2,1}(i) + I_\phi^{1,0}(i)$
(iii) $I_\phi^{r,s}(i) = I_h^{r,s}(i;\ (1/3, 1/3, 1/3))$

PROOF: The proofs of (i) and (ii) are trivial. To prove (iii), note that by summing over the 3^{n-1} possible values for X_k, $k \neq i$,

$$h(j_i,\ (1/3,1/3,1/3)) = 3^{-n+1} \sum_{\substack{\underline{x} \in S \\ x_i = j}} In_{\{\phi(j_i, \underline{x}) \geqslant m\}}$$

$$= 3^{-n} \sum_{\underline{x} \in S} In_{\{\phi(j_i, \underline{x}) \geqslant m\}}$$

where in the above, In_A denotes the indicator function of the set A. Thus

$$I_h^{2,1}(i;(1/3,1/3,1/3)) = 3^{-n} \sum_{\underline{x} \in S} [In_{\{\phi(2_i, \underline{x}) \geqslant m\}} - In_{\{\phi(1_i, \underline{x}) \geqslant m\}}]$$

$$= 3^{-n} n_\phi^{2,1}(i) = I_\phi^{2,1}(i).$$

The proof for $I_\phi^{1,0}(\cdot)$ is the same and the proof for $I_\phi^{2,0}(\cdot)$ follows from parts (i) and (ii). □

Parts (i) and (ii) of the above result show that both 2,0 importance measures decompose into the sum of the 2,1 and 1,0 importance measures. The generalized cut-importance ranking to be defined later has a similar property. In practice, it is likely that the 2,0 measures and rankings would be the most commonly used. However, the other measures and rankings can

be useful in providing more detailed information about which states are most relevant in determining a given component's ranking. (See Example 2.)

Given a generalized coherent system (N, ϕ), and a partition $\mathcal{C} = (C_0, C_1, C_2)$ of N into three sets, define $\underline{x}(\mathcal{C}) \in S$ by

$$(\underline{x}(\mathcal{C}))_i = \begin{cases} 0 & i \in C_0 \\ 1 & i \in C_1 \\ 2 & i \in C_2. \end{cases}$$

The function $\underline{x}(\mathcal{C})$ shows how any partition $\mathcal{C}$ determines the states of all the components.

DEFINITION: A partition $\mathcal{C} = (C_0, C_1, C_2)$ of N is a *cut* if and only if $\phi(\underline{x}(\mathcal{C})) < m$. A cut $\mathcal{C}$ is a *minimal cut* if and only if $\phi(\underline{y}) \geqslant m$ for all $\underline{y} \in S$ such that $\underline{y} \geqslant \underline{x}(\mathcal{C})$, $\underline{y} \neq \underline{x}(\mathcal{C})$.

While it is in principle possible to develop a complete cut-importance ranking for generalized coherent systems, in practice the calculation of the entire generalized $\underline{b}^{(\cdot)}$ vector for each component is too complex to be feasible. However, a partial ordering of the components which involves very few calculations can be developed by generalizing Proposition 1 and Corollary 1 appropriately. First, the notions of the size of a partition and the union of partitions must be defined.

DEFINITION: The *size* of a partition $\mathcal{C} = (C_0, C_1, C_2)$, denoted by $z(\mathcal{C})$, is $\alpha_0|C_0| + \alpha_1|C_1|$. ($\alpha_0$, α_1 are arbitrary constants satisfying $\alpha_0 > \alpha_1 > 0$.) The roles of the constants α_0, α_1 are dicussed later.

DEFINITION: Let $\mathcal{T}^1, \mathcal{T}^2, \ldots, \mathcal{T}^j$ be partitions of N where

$$\mathcal{T}^i = (T_0^i, T_1^i, T_2^i).$$

The *union* of $\mathcal{T}^1, \ldots, \mathcal{T}^j$ is the partition

$$(V_0, V_1 - V_0, V_2 - V_1 - V_0)$$

where $V_k = \bigcup_{i=1}^{j} T_k^i$.

DEFINITION: Consider a ternary coherent system with minimal cuts $\mathcal{C}^i = (C_0^i, C_1^i, C_2^i)$, $i = 1, 2, \ldots, t$. For each component k, let

$$e_k^{r+1,r} = \min_{1 \leqslant i \leqslant t} \{z(\mathcal{C}^i) : k \in C_r^i\} - \alpha_r,$$

and let

$$f_k^{r+1,r} = |\{\mathcal{C}^i : k \in C_r^i, z(\mathcal{C}^i) - \alpha_r = e_k^{r+1,r},\ 1 \leqslant i \leqslant t\}|.$$

(By convention $e_k^{r+1,r} = +\infty$ if $k \notin C_r^i, 1 \leqslant i \leqslant t$; $e_k^{r+1,r}$ is just the size of the "smallest" min cut which contains k in the r^{th} set of the partition less α_r, and $f_k^{r+1,r}$ is the number of such min cuts.) For all $r,s = 0,1,2$ such that $r > s$, define

$$e_k^{r,s} = \min_{s \leqslant u < r} \{e_k^{u+1,u}\}$$

and

$$f_k^{r,s} = \sum_{\substack{s \leqslant u < r \\ e_k^{u+1,u} = e_k^{r,s}}} f_k^{u+1,u}.$$

Component l is *more r,s cut-important* than component k, denoted $l >_c^{r,s} k$, if and only if either

(i) $e_l^{r,s} < e_k^{r,s}$,

or

(ii) $e_l^{r,s} = e_k^{r,s}$ and $f_l^{r,s} > f_k^{r,s}$.

The 2,0 cut-importance ranking will sometimes be simply called the *cut-importance ranking* and be denoted by $>_c$. As in the binary case, each of the r,s importance rankings is consistent with the ranking induced by the corresponding r,s importance measure. To be more specific, let $\underline{p}(\epsilon) = (\epsilon^{\alpha_0}, \epsilon^{\alpha_1}, 1 - \epsilon^{\alpha_0} - \epsilon^{\alpha_1})$. As ϵ approaches zero, $\underline{p}(\epsilon)$ puts almost all its mass on the best state, state 2. Of the mass left over, the ratio of the mass put on state 0 to that put on state 1 approaches zero. Thus the parameters α_0 and α_1 give the relative weights put on components in state zero versus components in state 1 in the cut importance ranking and also determine the relative likelihoods of a component partially failing (state 1) and fully failing (state zero).

THEOREM 3: For ϵ sufficiently close to zero, the component ranking induced by $I_h^{r,s}(\cdot;\underline{p}(\epsilon))$ is consistent with the r,s cut-importance ranking ($r > s$).

PROOF: $I_h^{r+1,r}(k;\underline{p}(\epsilon)) = h((r+1)_k, \underline{p}(\epsilon)) - h(r_k, \underline{p}(\epsilon))$

$$= 1 - Pr\left\{\bigcup_{i=1}^{t} E_i | X_k = r+1\right\} - \left[1 - Pr\left\{\bigcup_{i=1}^{t} E_i | X_k = r\right\}\right],$$

where $E_i = \{\underline{X} \leqslant \underline{x}(\mathcal{C}^i)\}$. Thus by the inclusion-exclusion principle,

$$I_h^{r+1,r}(k;\underline{p}(\epsilon)) = \sum_{i=1}^{t} (-1)^{i-1} [S_i^r - S_i^{r+1}], \tag{1}$$

where

$$S_i^r - S_i^{r+1} = \sum_{\underline{j} \in J_i} \left[Pr\left\{\bigcap_{l=1}^{i} E_{j_l} | X_k = r\right\} - Pr\left\{\bigcap_{l=1}^{i} E_{j_l} | X_k = r+1\right\}\right]$$

and $J_i = \{(j_1, j_2, \ldots, j_i): 1 \leqslant j_1 < j_2 < \cdots < j_i \leqslant t\}$. Letting $G = \{\underline{j} \in J_i: \min_{1 \leqslant l \leqslant i} \{X_k(\mathcal{C}^{j_l})\} = r\}$,

$$S_i^r - S_i^{r+1} = \sum_{\underline{j} \in G} Pr\left\{\bigcap_{l=1}^{i} E_{j_l} | X_k = r\right\}$$

because the two probabilities in the first expression for $S_i^r - S_i^{r+1}$ are equal for all $\underline{j} \in J_i - G$ since the X_j's are independent) and for $\underline{j} \in G$ the second probability is zero.

$$S_i^r - S_i^{r+1} = \sum_{\underline{j} \in G} Pr\left\{X_w \leqslant \min_{1 \leqslant l \leqslant i} \{X_w(\mathcal{C}^{j_l})\},\ 1 \leqslant w \leqslant n | X_k = r\right\}$$

$$= \sum_{\underline{j} \in H} Pr\{\underline{X} \leqslant \underline{x}(\mathcal{D}) | X_k = r\}$$

$$= \sum_{\underline{j} \in H} \epsilon^{-\alpha_r} \cdot \epsilon^{\alpha_0 |D_0|} \cdot (\epsilon^{\alpha_0} + \epsilon^{\alpha_1})^{|D_1|}$$

where $H = \{j \in J_i: k \in D_r\}$ and where $\mathcal{D} = (D_0, D_1, D_2)$ is the union of $\mathcal{C}^{j_1}, \mathcal{C}^{j_2}, \ldots, \mathcal{C}^{j_i}$. By the definitions of e_k and f_k, the lowest order term in this polynomial expression for $S_1^r - S_1^{r+1}$ is $f_k(\epsilon)^{e_k}$. [For notational simplicity the superscript $r + l$, r which should appear on e_k, f_k, $>_c$, and I_h will be dropped in the remainder of the proof.] Thus

$$S_1^r - S_1^{r+1} = f_k \epsilon^{e_k} + o(\epsilon^{e_k}).$$

Next we show that $S_i^r - S_i^{r+1} = o(\epsilon^{e_k})$ for all $i \geqslant 2$. Let $\mathcal{D}$ be the union of any i minimal cuts $\mathcal{C}^{j_1}, \ldots, \mathcal{C}^{j_i}$ satisfying $k \in D_r$. Then

$$|D_0| \geqslant |C_0^{j_1}|$$

and

$$|D_0| + |D_1| \geqslant |C_0^{j_1}| + |C_1^{j_1}|. \tag{2}$$

Assume that the above two inequalities simultaneously hold as equalities. Then $|D_0| = |C_0^{j_1}|$, which implies that $D_0 = C_0^{j_1}$ and $|D_1| = |C_1^{j_1}|$, which implies that $D_1 = C_1^{j_1}$. Thus $\mathcal{D} = \mathcal{C}^{j_1}$ and so $\underline{x}(\mathcal{D}) = \underline{x}(\mathcal{C}^{j_1})$. Now as an immediate consequence of the definition of $\mathcal{D}$, $\underline{x}(\mathcal{D}) \leqslant \underline{x}(\mathcal{C}^{j_2})$, and so $\underline{x}(\mathcal{C}^{j_1}) \leqslant \underline{x}(\mathcal{C}^{j_2})$. Furthermore, $\mathcal{C}^{j_1} \neq \mathcal{C}^{j_2}$, so the inequality must be strict in at least one coordinate. But this contradicts the assumption that the cut $\mathcal{C}^{j_1}$ is minimal, and so at least one of the inequalities in (2) must be strict.

Now the lowest power in the polynomial expression for $S_i^r - S_i^{r+1}$ is $\alpha_0|D_0| + \alpha_1|D_1| - \alpha_r$. But

$$\begin{aligned}\alpha_0|D_0| + \alpha_1|D_1| - \alpha_r &= (\alpha_0 - \alpha_1)|D_0| + \alpha_1(|D_0| + |D_1|) - \alpha_r \\ &> (\alpha_0 - \alpha_1)|C_0^{j_1}| + \alpha_1(|C_0^{j_1}| + |C_1^{j_1}|) = \alpha_r \\ &> \alpha_0|C_0^{j_1}| + \alpha_1|C_1^{j_1}| - \alpha_r \geqslant e_k.\end{aligned}$$

Thus

$$S_i^r - S_i^{r+1} = o(\epsilon^{e_k}).$$

Thus, by equation (1)

$$I_h(k;\underline{p}(\epsilon)) = f_k \cdot \epsilon^{e_k} + o(\epsilon^{e_k}). \tag{3}$$

Now assume that $l >_c k$. If $e_l < e_k$, then by equation (3)

$$I_h(l;\underline{p}(\epsilon)) - I_h(k;\underline{p}(\epsilon)) = f_l \cdot \epsilon^{e_l} + o(\epsilon^{e_l}).$$

Thus for ϵ sufficiently close to zero this expression is positive and so the two orderings of l and k are identical in this case.

If $e_l = e_k$ and $f_l > f_k$, then again by equation (3)

$$I_h(l;\underline{p}(\epsilon)) - I_h(k;\underline{p}(\epsilon)) = (f_l - f_k)\epsilon^{e_l} + o(\epsilon^{e_l}).$$

Thus in this case, also, the two orderings are consistent for ϵ sufficiently small. This establishes the theorem for all the $r+1$, r orderings. To establish the result for any r, s ordering note that

$$I_h^{r,s}(k;\underline{p}(\epsilon)) = \sum_{u=s}^{r-1} I_h^{u+1,u}(k;\underline{p}(\epsilon)).$$

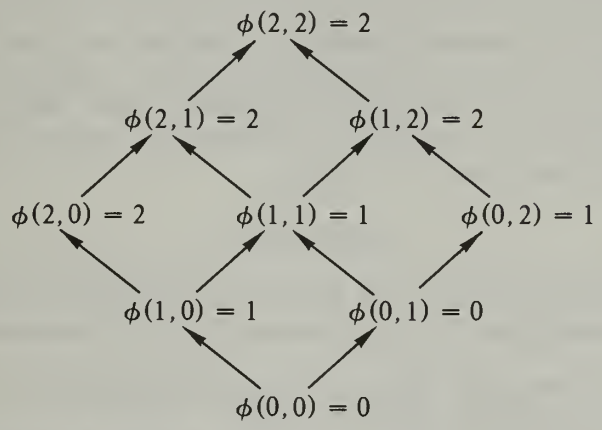
(See Proposition 3, part (i).) Combining this result with equation (3),

$$I_h^{r,s}(k;\underline{p}(\epsilon)) = f_k^{r,s} \epsilon^{e_k^{r,s}} + o(\epsilon^{e_k^{r,s}}).$$

The remainder of the proof is identical to the $r+1$, r case. [*Note*: For ternary systems, the only $r+1$, r orderings are the 2-1 ordering and the 1-0 ordering. The only other r, s ordering is the 2-0 ordering. The notation in the proof and the definition of r, s cut-importance has been kept more general so that the extensions to general n-state systems can be more readily understood.]

□

EXAMPLE 2: In the following diagram, the states are shown in a lattice arrangement according to the less-than-or-equal-to relation.



Consider the case where $m=2$, $\alpha_0=2$, $\alpha_1=1$.

Min cuts: $\mathcal{C}^1=(E,\{1,2\},E)$ (E denotes the empty set.)

$\mathcal{C}^2=(\{1\},E,\{2\})$

$$\left.\begin{array}{l} e_1^{2,1}=1;\ e_2^{2,1}=1 \\ \\ f_1^{2,1}=1;\ f_2^{2,1}=1 \end{array}\right\} \Rightarrow \text{Components 1, 2 are not comparable under the 2, 1 ranking.}$$

$$\left.\begin{array}{l} e_1^{1,0}=0;\ e^{1,0}=+\infty \\ \\ f_1^{1,0}=1;\ f_2^{1,0}=0 \end{array}\right\} \Rightarrow 1 >_c^{1,0} 2$$

$$\left.\begin{array}{l} e_1^{2,0}=0;\ e_2^{2,0}=1 \\ \\ f_1^{2,0}=1;\ f_2^{2,0}=1 \end{array}\right\} \Rightarrow 1 >_c^{2,0} 2$$

$$h(P)=1-(p_{10}+p_{11})(p_{20}+p_{21})-p_{10}+p_{10}(p_{20}+p_{21})$$
$$=1-p_{11}(p_{20}+p_{21})-p_{10}.$$

$$I_h^{2,1}(1;\underline{p}(\epsilon))=\epsilon^2+\epsilon. \qquad I_h^{2,1}(2;\underline{p}(\epsilon))=\epsilon.$$
$$I_h^{1,0}(1;\underline{p}(\epsilon))=1-\epsilon-\epsilon^2. \qquad I_h^{1,0}(2;\underline{p}(\epsilon))=0.$$
$$I_h^{2,0}(1;\underline{p}(\epsilon))=1. \qquad I_h^{2,0}(2;\underline{p}(\epsilon))=\epsilon.$$

Thus component 1 is more important than component 2 in an overall sense (i.e., according to the 2, 0 cut-importance ranking). Moreover, the 2, 1 and 1, 0 rankings of the components

show that it is the state 1 to state 0 transition of the components which determines the 2, 0 ranking here.

As was the case for binary systems, analogous results based upon minimal paths can be developed for ternary systems composed of very unreliable components.

4. CONCLUSIONS

Reliability engineers usually know or can calculate the minimal cuts of the systems with which they deal. However, the component reliabilities, though usually thought to be fairly high, are often not known with any degree of precision. The cut-importance rankings developed in this paper are structural rankings, i.e., depend only upon the system structure through the minimal cuts. Also they are defined so as to relate closely to the Birnbaum reliability importance measure when the component reliabilities are high. Thus they provide reliability analysts and engineers with ways to meaningfully compare the relative importance to the system reliability of the various components of the system.

REFERENCES

[1] Barlow, R.E., "Coherent Systems with Multi-State Components," University of California Operations Research Center Technical Report ORC 77-5, Berkeley, California (January 1977).

[2] Barlow, R.E. and F. Proschan, *Statistical Theory of Reliability and Life Testing: Probability Models* (Holt, Rinehart, and Winston, 1975).

[3] Barlow, R.E. and F. Proschan, "Importance of System Components and Fault Tree Events," *Stochastic Processes and Their Applications*, Vol. 3, pp. 153-172 (1975).

[4] Birnbaum, Z.W., "On the Importance of Different Components in a Multi-Component System," in *Multivariate Analysis—II*, P.R. Krishnaiah (ed.) (Academic Press, New York, 1969).

[5] Butler, D.A., "An Importance Ranking for System Components Based upon Cuts," Operations Research, Vol. 25, No. 5, pp. 874-879 (1977).

[6] Butler, D.A., "A Complete Importance Ranking for Components of Binary Coherent Systems, with Extensions to Multi-State Systems," Technical Report No. 183, Department of Operations Research, Stanford University, Stanford, CA (1977).

[7] El-Meweihi, E., F. Proschan and J. Sethuraman, "Multi-State Coherent Systems," Florida State University Statistics Report M434 (October 1977).

[8] Feller, W., *An Introduction to Probability Theory and its Applications*, Vol. I, 3rd ed., pp. 98-101 (John Wiley and Sons, New York, 1968).

[9] Hatoyama, Y., "Fundamental Concepts for Reliability Analysis of Three-State Systems," unpublished manuscript, Department of Operations Research, Stanford University (1976).

[10] Lambert, H.E., "Measures of Importance of Events and Cut-Sets in Fault Trees," Lawrence Livermore Laboratory UCRL-75853 (October 1974).

[11] Murchland, J.D., "Fundamental Concepts and Relations for Reliability Analysis of Multi-State Systems," *Reliability and Fault Tree Analysis,* Society for Industrial and Applied Mathematics (1975).

[12] Postelnicu, V., "Nondichotomic Multi-Component Structures," Bulletin Mathematique de la Societe des Sciences Mathematiques de la Republique Socialiste de Roumanie, Vol. 14, (62), No 2, pp. 209-217, (1970).

A DIFFUSION MODEL FOR THE CONTROL OF A MULTIPURPOSE RESERVOIR SYSTEM

Dror Zuckerman*

Department of Operations Research
College of Engineering
Cornell University
Ithaca, New York

ABSTRACT

This paper develops a methodology for optimizing operation of a multipurpose reservoir with a finite capacity V. The input of water into the reservoir is a Wiener process with positive drift. There are n purposes for which water is demanded. Water may be released from the reservoir at any rate, and the release rate can be increased or decreased instantaneously with zero cost. In addition to the reservoir, a supplementary source of water can supply an unlimited amount of water demanded during any period of time. There is a cost of C_i dollars per unit of demand supplied by the supplementary source to the i^{th} purpose ($i = 1, 2, \ldots, n$). At any time, the demand rate R_i associated with the i^{th} purpose ($i = 1, 2, \ldots, n$) *must be* supplied. A controller must continually decide the amount of water to be supplied by the reservoir for each purpose, while the remaining demand will be supplied through the supplementary source with the appropriate costs. We consider the problem of specifying an output policy which minimizes the long run average cost per unit time.

. INTRODUCTION AND FORMULATION

Complex systems of reservoirs today are used to produce supplies of water for agriculture, ıdustry and urban use. In addition, the production of hydroelectric power is usually a major bjective of water resource systems.

An excellent account of the theory of storage systems, describing results obtained up to 964 is contained in Prabhu's paper [9]. Considerable progress has since been made in several rections, but most of the models are descriptive, rather than control-oriented. Dynamic pro-amming models for the optimal control of multipurpose reservoir systems have been pro-ɔsed by Hall, Butcher and Esogbue [4], Russell [10] and many others. Most of the models volved discrete time analysis. Meanwhile other authors, notably Bather [1], Faddy [2], [3], aslett [5] and Pliska [8], have developed diffusion models for the control of a dam with finite servoir capacity, where the optimality was defined in terms of a cost (or a utility) structure ıposed on the operation of the system. The main purpose of this article is to provide an addi-ɔnal insight into the nature of the optimal controls.

---

low at The Hebrew University of Jerusalem.

iis research was supported in part by the National Science Foundation under Grant MBS 73-04437.

In the present study we develop a methodology for optimizing operation of a multipurpose reservoir with a finite capacity V described by the following model: The input of water into the reservoir is determined by a Wiener process with positive drift μ and variance σ^2. There are n purposes for which water is demanded. Water may be released from the reservoir at any rate R, $(R \geqslant 0)$. Let R_i be the demand in units of water per unit time associated with the i^{th} purpose $(i = 1, 2, \ldots, n)$. At any time the release rate may be increased or decreased with zero cost, any such changes taking effect instantaneously. In addition to the reservoir, a supplementary source of water can supply an unlimited amount of water demanded during any period of time. There is a cost of C_i dollars per unit of demand provided by the supplementary source to the i^{th} purpose, $(i = 1, 2, \ldots, n)$. We assume without loss of generality that $C_1 \geqslant C_2 \geqslant \ldots \geqslant C_n$. At any time, the demand rate R_i associated with the i^{th} purpose $(i = 1, 2, \ldots, n)$ *must be* supplied. A controller must continually decide the amount of water to be supplied by the reservoir for each purpose, while the remaining demand will be supplied through the supplementary source with the appropriate costs. We consider the problem of specifying an output policy which minimizes the long run average cost per unit time. An example will be presented to illustrate computational procedures.

2. THE MODEL

Let us denote by $X(t)$ the input into the dam during the time interval (0,t]; as indicated earlier $\{X(t); t \geqslant 0\}$ is a Wiener process. By an appropriate choice of units, we may assume without loss of generality that $\mu = 1$, $\sigma^2 = 2$.

Note that negative values of the storage level (as in [1]) have to be taken into account. This representation is relatively crude, but a solution to the problem of optimal control is still useful, since control is needed only when the storage level is positive and we necessarily have for non-positive values of the storage level process that the demand associated with the n purposes will be supplied totally through the supplementary source, under any permissible output policy.

If we assume, as in Pliska's paper [8] that 0 is a reflecting boundary, then the expected time that the dam is empty (dry) over any given period is 0, independently of the output policy which is employed. The above situation seems to be unrealistic for most reservoir models. Furthermore, the multipurpose reservoir model which is considered by us becomes meaningless, since the optimal policy in this case is to supply the demand associated with the n purposes *totally* through the reservoir and the *resulting average cost associated with the above policy will be zero.* In view of this, the Bather model in our case seems to be more appropriate.

It will be very helpful for us to restrict the state space of the storage level process to a finite interval (in order to apply some results obtained by Mandl [6] and Pliska [7]). Therefore we make the following modification: Assume that the storage level is bounded from below by -1, where -1 is an elementary return boundary. That is, when the trajectory of the storage level process reaches the boundary -1 it remains at -1 for a random amount of time which possesses the exponential distribution with mean 1, and after the termination of the sojourn time on the boundary -1, the process jumps into position 0 with probability 1. Clearly, the expected transition time from -1 to 0 is the same as under the original process. Thus, since our goal is to minimize the total long run average cost per unit time, the above modification of the storage level process does not affect the decision problem (it is just a *mathematical tool*).

Now let us consider the set of admissible output policies. In selecting the output policy only the current state, that is level of water in the reservoir, is important. The particular time

s irrelevant since the input process is time homogeneous and since we are concerned with an nfinite future. Thus we consider only policies Γ such that

$$\Gamma;\ [-1,\ V] \to S$$

vhere

$$S = \{(\alpha_1,\ \alpha_2,\ \ldots,\ \alpha_n)\ |0 \leqslant \alpha_i \leqslant 1, \quad \text{for } i = 1, 2,\ \ldots,\ n\}.$$

n words, $\Gamma(x) = (\Gamma_1(x),\ \Gamma_2(x),\ \ldots,\ \Gamma_n(x))$ means that under an output policy Γ, when the torage level is x, $100\Gamma_i(x)$ percent of the demand rate associated with the i^{th} purpose will be upplied by the reservoir, and $100(1 - \Gamma_i(x))$ percent of it will be provided by the supplementary source, for $i = 1,\ 2,\ \ldots,\ n$.

The set M of admissible controls consists of all piecewise continuous functions $\Gamma(\cdot)$ on $-1,\ V]$ with range in S.

Let $\{Z_\Gamma;\ \Gamma \in M\}$ be the controlled process, corresponding to the storage level in the dam vhen a policy $\Gamma \in M$ is employed.

The controlled process $\{Z_\Gamma;\ \Gamma \in M\}$ is a diffusion process whose state space is the interal $(-1,\ V)$, with drift parameter

1)
$$b_\Gamma(x) = 1 - \sum_{i=1}^{n} \Gamma_i(x) R_i$$

nd diffusion parameter (see Mandl [6], p. 12)

2)
$$a_\Gamma(x) = \frac{\sigma^2}{2} = 1.$$

Ve assume that V is a reflecting boundary.

The cost arising from continuous movement of the controlled process is given by a ounded piecewise continuous function $C_\Gamma(x) = \sum_{i=1}^{n} C_i R_i\ (1 - \Gamma_i(x))$ defined on $[-1,\ V]$. If ıe trajectory is in position x at time t, then there arises a cost of the magnitude $C_\Gamma(x)\Delta t +$ (Δt) in the time interval $(t,\ t + \Delta t)$.

We want to find a control $\Gamma^* \in M$ which minimizes the long run average cost per unit me.

THE OPTIMAL POLICY

In this section we show that the optimal output policy has the following form

)
$$\Gamma(z) = \begin{cases} (0,0,0,\ \ldots,\ 0,0) & \text{if } z = 0 \\ (1,0,0,\ \ldots,\ 0,0) & \text{if } 0 < z \leqslant \gamma_1 \\ (1,1,0,\ \ldots,\ 0,0) & \text{if } \gamma_1 < z \leqslant \gamma_2 \\ (1,1,1,0,\ \ldots,\ 0) & \text{if } \gamma_2 < z \leqslant \gamma_3 \\ \quad\vdots & \\ (1,1,1,\ \ldots,\ 1,1) & \text{if } z > \gamma_{n-1} \end{cases}$$

where

$$0 \leqslant \gamma_1 \leqslant \gamma_2 \leqslant \ldots \leqslant \gamma_{n-1} \leqslant V.$$

(Recall that $C_1 \geqslant C_2 \geqslant \ldots \geqslant C_n$ by assumption).

In the special case in which $C_1 = C_2 = \ldots = C_n$, as one might anticipate, the optimal control would specify the maximum discharge rate at all positive levels, i.e., $\gamma_1^* = \gamma_2^* = \ldots = \gamma_{n-1}^* = 0$.

First consider the subclass of policies $L \subset M$ under which, at any given state, the demand associated with the i^{th} purpose $(i = 1, 2, \ldots, n)$, will be satisfied totally by the reservoir, or totally by the supplementary source. It can easily be seen that L is the collection of all piecewise constant functions $\Gamma(\cdot)$ on $[-1, V]$ with range in $\hat{S} \subset S$, where

$$\hat{S} = \{(\alpha_1, \alpha_2, \ldots, \alpha_n) \mid \alpha_i = 0 \text{ or } 1, \text{ for } i = 1, 2, \ldots, n\}.$$

Thus under an output policy $\Gamma \in L$, the set $(0, V]$ can be decomposed into a finite number of intervals, on each of which Γ is a constant. That is, for each output policy $\Gamma \in L$, there exists a real sequence $\{\gamma_i\}_{i=0}^{p}$ such that

$$0 = \gamma_0 < \gamma_1 < \gamma_2 < \ldots < \gamma_p = V,$$

where $\Gamma(\cdot)$ is constant over each interval (γ_{i-1}, γ_i) for $i = 1, 2, \ldots, p$. For each policy $\Gamma \in L$, we define the following sets:

$$A_i(\Gamma) = \{j \mid 1 \leqslant j \leqslant n,\ \Gamma_j(x) = 1 \text{ for } x \in (\gamma_{i-1}, \gamma_i)\},\ (1 \leqslant i \leqslant p). \tag{4}$$

The theory of optimal control in diffusion processes (Theorem 5 of Mandl [6], p. 168) implies that under a given policy $\Gamma \in M$, the long run average cost per unit time, ϕ_Γ, is the *unique* number to which there exists a continuous function $w_\Gamma(\cdot)$ on $[-1, V]$ such that

$$\frac{dw_\Gamma(z)}{dz} + b_\Gamma(z) w_\Gamma(z) + C_\Gamma(z) - \phi_\Gamma = 0 \tag{5}$$

holds for every $z \in (-1, V)$ which is a point of continuity of Γ, and that

$$w_\Gamma(0) = \phi_\Gamma - \sum_{j=1}^{n} R_j C_j, \tag{6}$$

$$w_\Gamma(V) = 0. \tag{7}$$

Let

$$\alpha_i(\Gamma) = 1 - \sum_{j \in A_i(\Gamma)} R_j$$

and

$$\beta_i(\Gamma) = \sum_{j \notin A_i(\Gamma)} R_j C_j.$$

The general solution of the differential equation (5), assuming $\alpha_i(\Gamma) \neq 0$ for a policy $\Gamma \in L$ is given by

$$w_\Gamma(z) = \frac{\phi_\Gamma - C_\Gamma(z)}{b_\Gamma(z)} + \frac{e^{-zb_\Gamma(z)}}{b_\Gamma(z)} d_i = \frac{\phi_\Gamma - \beta_i(\Gamma)}{\alpha_i(\Gamma)} + \frac{e^{-\alpha_i(\Gamma) z}}{\alpha_i(\Gamma)} d_i, \tag{8}$$

for $z \in (\gamma_{i-1}, \gamma_i)$, $(i = 1, 2, \ldots, p)$, where d_i are arbitrary constants. The solution for the case $\alpha_i(\Gamma) = 0$ can be obtained by considering the limiting behaviour of (8) as $\alpha_i(\Gamma)$ approaches 0.

Recalling that $w_\Gamma(\cdot)$ is a continuous function over $[-1, V]$, the constants d_i must be hosen to assure continuity of $w_\Gamma(\cdot)$ at the points γ_i $(i = 1, 2, \ldots, p-1)$. Thus we obtain the ollowing equations:

$$\frac{\phi_\Gamma - \beta_i(\Gamma)}{\alpha_i(\Gamma)} + \frac{e^{-\alpha_i(\Gamma)\gamma_i}}{\alpha_i(\Gamma)} d_i = \frac{\phi_\Gamma - \beta_{i+1}(\Gamma)}{\alpha_{i+1}(\Gamma)} + \frac{e^{-\alpha_{i+1}(\Gamma)\gamma_i}}{\alpha_{i+1}(\Gamma)} d_{i+1} \tag{9}$$

or $i = 1, 2, \ldots, p-1$. The optimal control Γ^* will be determined with the aid of the follow- ng Theorem.

THEOREM 1: A control Γ^* is optimal if and only if

$$\Gamma_i^*(x) = \begin{cases} I_{(x>0)} & \text{if } C_i + w_{\Gamma^*}(x) \geqslant 0 \\ 0 & \text{if } C_i + w_{\Gamma^*}(x) < 0 \end{cases} \tag{10}$$

or $i = 1, 2, \ldots, n$, and for every x which is a continuity point of Γ^*, where w_{Γ^*} is the solu- on of the differential equation (5) when policy Γ^* is employed, and $I_{(E)}$ is the indicator func- on of the event E.

PROOF: Let

$$\theta_{\Gamma,\psi}(x) = b_\psi(x)\ w_\Gamma(x) + C_\psi(x), \text{ for } \Gamma \in M \text{ and } \psi \in M.$$

ccording to Theorem 6 of Mandl ([6], p. 168), Γ^* is optimal if and only if

$$\theta_{\Gamma^*,\Gamma^*}(x) = \min_{\psi \in M} \{\theta_{\Gamma^*,\psi}(x)\} \tag{11}$$

r every $x \in (0, V]$ which is a continuity point of Γ^*. (Note that $a_\psi(x) = 1$ for every position $\in (0, V]$, under any output policy $\psi \in M$). For a given policy $\psi \in M$ we have

$$\begin{aligned} \theta_{\Gamma^*,\psi}(x) &= (1 - \sum_{i=1}^{n} R_i\psi_i(x))\ w_{\Gamma^*}(x) + \sum_{i=1}^{n} (1 - \psi_i(x))\ C_iR_i \\ &= w_{\Gamma^*}(x) + \sum_{i=1}^{n} C_iR_i - \sum_{i=1}^{n} R_i\psi_i(x)\ [w_{\Gamma^*}(x) + C_i]. \end{aligned} \tag{12}$$

then clearly follows that $\theta_{\Gamma^*,\Gamma^*}(x) = \min_{\psi} \{\theta_{\Gamma^*,\psi}(x)\}$ for every $x \in (0, V]$ which is a con- nuity point of Γ^*, if and only if (10) holds for $i = 1, 2, \ldots, n$. This concludes the proof. □

Generally, there is no guarantee that an admissible optimal control will exist. However, i our case, it follows from Theorem 1 that if an optimal output policy $\Gamma^* \in M$ exists, then $^* \in L$. But the existence of an optimal control in the subclass L of policies follows directly om Theorem 4.1 of Pliska [7]. Thus there exists an optimal control in M. We proceed with te following proposition.

PROPOSITION 1: Let Γ^* be the optimal output policy, then w_{Γ^*} is nondecreasing over te interval $[0, V]$.

PROOF: We will summarize briefly the main steps of the proof. Clearly $\cdot - \sum_{i=1}^{n} R_iC_i \leqslant 0$. Using equations (6) and (7), we obtain the following inequality

$$w_{\Gamma^*}(0) \leqslant w_{\Gamma^*}(V) = 0. \tag{13}$$

From (13) it can be seen by elementary analysis that if $w_{\Gamma^\bullet}$ is *not* nondecreasing, then there exist two points x and y in the open interval $(0, V)$, which are continuity points of Γ^*, such that

$$w_{\Gamma^\bullet}(x) = w_{\Gamma^\bullet}(y) \tag{14}$$

and

$$\left.\frac{dw_{\Gamma^\bullet}(z)}{dz}\right|_{z=x} < 0, \tag{15a}$$

$$\left.\frac{dw_{\Gamma^\bullet}(z)}{dz}\right|_{z=y} > 0. \tag{15b}$$

Using equations (10) and (14) we obtain that $\Gamma^*(x) = \Gamma^*(y)$. Hence,

$$b_{\Gamma^\bullet}(x) = b_{\Gamma^\bullet}(y) \tag{16}$$

and

$$C_{\Gamma^\bullet}(x) = C_{\Gamma^\bullet}(y). \tag{17}$$

From (5) we obtain

$$\left.\frac{dw_{\Gamma^\bullet}(z)}{dz}\right|_{z=x} + b_{\Gamma^\bullet}(x)\, w_{\Gamma^\bullet}(x) - \phi_{\Gamma^\bullet} + C_{\Gamma^\bullet}(x) = \left.\frac{dw_{\Gamma^\bullet}(z)}{dz}\right|_{z=y} + b_{\Gamma^\bullet}(y)\, w_{\Gamma^\bullet}(y) - \phi_{\Gamma^\bullet} + C_{\Gamma^\bullet}(y), \tag{18}$$

now substituting equations (14), (16) and (17) into (18) we have

$$\left.\frac{dw_{\Gamma^\bullet}(z)}{dz}\right|_{z=x} = \left.\frac{dw_{\Gamma^\bullet}(z)}{dz}\right|_{z=y}$$

which is a contradiction to (15a) and (15b). Hence $w_{\Gamma^\bullet}$ is nondecreasing. □

Recalling that $C_1 \geqslant C_2 \geqslant \ldots \geqslant C_n$, and by using proposition 1 and Theorem 1, it follows that if $\Gamma_j^*(x) = 1$ for a given j, then $\Gamma_i^*(z) = 1$ for z such that $x \leqslant z \leqslant V$ and for each i such that $1 \leqslant i \leqslant j$. Now in order to establish that the optimal output policy has the form given in (3), we still have to show that $\Gamma_1^*(x) = 1$ for every positive x. But this is a direct consequence of the following Proposition.

PROPOSITION 2: The optimal output policy Γ^* satisfies the following condition

$$w_{\Gamma^\bullet}(0) + C_1 \geqslant 0.$$

PROOF: The proof will be done by contradiction. Suppose that

$$w_{\Gamma^\bullet}(0) + C_1 = -\delta < 0 \tag{19}$$

Since $w_{\Gamma^\bullet}$ is continuous on $[0, V]$ it follows that there exists $\epsilon(\delta) > 0$ such that $w_{\Gamma^\bullet}(y) + C_1 < 0$ for $0 \leqslant y \leqslant \epsilon(\delta)$.

Now recalling that $w_{\Gamma^\bullet}(\cdot)$ satisfies (10), it follows that $\Gamma_i^*(y) = 0$ for $0 \leqslant y \leqslant \epsilon(\delta)$ and for $i = 1, 2, \ldots, n$. Using equations (6) and (8) we have

$$w_{\Gamma^\bullet}(y) = \phi_{\Gamma^\bullet} - \sum_{i=1}^{n} C_i R_i \text{ for } 0 \leqslant y \leqslant \epsilon(\delta).$$

nce $w_{\Gamma^*}(0) = w_{\Gamma^*}(\epsilon(\delta))$, we can repeat the same argument over the intervals $[\epsilon(\delta), \epsilon(\delta)]$ ... and therefore $\Gamma^*(y) = 0$ for $i = 1, 2, \ldots, n$ and for every position $y \in [0, V]$. ius Γ^* is the trivial policy that keeps the output rate constantly at zero. But (7) must hold, so $\cdot = \sum_{i=1}^{n} C_i R_i$ and $w_{\Gamma^*}(0) = 0$. But $C_1 > 0$, so

0)
$$w_{\Gamma^*}(0) + C_1 = \phi_{\Gamma^*} - \sum_{i=1}^{n} C_i R_i + C_1 > 0,$$

hich is a contradiction to (19). Therefore

$$w_{\Gamma^*}(0) + C_1 \geqslant 0$$

required.

□

This concludes the proof that the optimal output policy is of the form given in (3), as sired.

In the following example, we will illustrate how to determine the optimal control values $\gamma_2^*, \ldots, \gamma_{n-1}^*$.

EXAMPLE: Let us consider the following case: We have a finite dam with capacity of $= 100$. There are two types of demand for water, where

$R_1 = 0.9$, $R_2 = 0.2$,
$C_1 = KC$, $(K \geqslant 1)$, $C_2 = C$.

st note that for a given policy $\Gamma \in L$ which has the form of (3), w_Γ is given by (see (8))

1)
$$w_\Gamma = \begin{cases} 10_{\phi_\Gamma} - 2C + 10e^{-0.1z}d_1 & \text{for } 0 \leqslant z < \gamma_1 \\ -10_{\phi_\Gamma} - 10e^{0.1z}d_2 & \text{for } \gamma_1 \leqslant z \leqslant V = 100. \end{cases}$$

ing (6), (7) and (9) we obtain the following equations

1. $10\phi_\Gamma - 2C + 10d_1 = \phi_\Gamma - C(0.9K + 0.2)$
2. $-10\phi_\Gamma - 10e^{10}d_2 = 0$
3. $10\phi_\Gamma - 2C + 10e^{-0.1\gamma_1}d_1 = -10\phi_\Gamma - 10e^{0.1\gamma_1}d_2$.

From the above 3 equations we obtain that the long run average cost associated with an put policy Γ, which has the form given in (3) will be

!)
$$\phi_\Gamma = \frac{2 - 0.9\delta^{-1}(2-K)}{20 - 10\delta e^{-10} - 9\delta^{-1}} C,$$

ere $\delta = e^{0.1\gamma_1}$. From the cost function introduced above it can be seen that only the ratio of ts $K = \dfrac{C_1}{C_2}$ is important in order to determine the optimal critical value γ_1^*.

Suppose that $K = 4$, then using (22) one can easily obtain that $\gamma_1^* \approx 55$ minimizes ϕ_Γ ject to the following constraint

$$0 \leqslant \gamma_1 \leqslant V = 100,$$

a $\phi_{\Gamma^*} \approx 0.1C$.

ACKNOWLEDGMENT

The author acknowledges helpful and illuninating discussions with Professor N.U. Prabhu.

REFERENCES

[1] Bather, J., "A Diffusion Model for the Control of a Dam," Journal of Applied Probability, *5*, 55-71 (1968).

[2] Faddy, M.J., "Optimal Control of Finite Dams: Continuous Output Procedure," Advances in Applied Probability, *6*, 689-710 (1974).

[3] Faddy, M.J., "Optimal control of finite dams: Discrete (2-stage) output procedure," Journal of Applied Probability *11*, 111-121 (1974).

[4] Hall, W.A., W.S. Butcher and A.M.O. Esogbue, "Optimization of the operation of a multi-purpose reservoir by dynamic programming," Water Resources Research *4*, 471-477 (1968).

[5] Haslett, J., "The control of a multi-purpose reservoir," Advances in Applied Probability *8*, 592-609 (1976).

[6] Mandl, P., *Analytical Treatment of One-Dimensional Markov Processes.* (Springer-Verlag, New York 1968).

[7] Pliska, S.R., "Single person controlled diffusions with discounted costs," Journal of Optimization Theory and Application *12*, 248-255 (1973).

[8] Pliska, S.R., "A diffusion process model for the optimal operation of a reservoir system," Journal of Applied Probability *12*, 859-863 (1975).

[9] Prabhu, N.U., "Time-dependent results in storage theory," Journal of Applied Probability *1*, 1-46 (1964).

[10] Russel, C.B., "An optimal policy for operating a multipurpose reservoir," Operations Research *20*, 1181-1189 (1972).

COMPUTATION TECHNIQUES FOR LARGE SCALE UNDISCOUNTED MARKOV DECISION PROCESSES

Thom J. Hodgson and Gary J. Koehler

University of Florida
Gainesville, Florida

ABSTRACT

In this paper we consider computation techniques associated with the optimization of large scale Markov decision processes. Markov decision processes and the successive approximation procedure of White are described. Then a procedure for scaling continuous time and renewal processes so that they are amenable to the White procedure is discussed. The effect of the scale factor value on the convergence rate of the procedure and insights into proper scale factor selection are given.

INTRODUCTION

One of the most powerful modeling tools for the analysis of controlled probabilistic systems is Markov decision processes. If the system can be structured as a Markov process and the control decisions for the system can be defined in terms of the relevant system costs and operational characteristics (transition probabilities), then there exists a wealth of theory that can be used to find the best (least cost, most profitable) set of decisions for operating the system. As with many modeling techniques, real probabilistic systems, when modeled as Markov processes, tend to have large numbers of system states. The result is that for many interesting and important systems, it is necessary to consider the computational aspects associated with performing policy optimization.

Many types of nondiscounted Markov decision processes can be transformed to a discrete time problem. Such a procedure was explicitly used by Schweitzer [17] for Markov renewal programs and involves choosing a parameter, b, for the transformation. As noted by Schweitzer, the value of b influences the asymptotic convergence rate when White's iterative procedure [22] is used to solve the transformed Markov decision process. We present theoretical insights into the determination of a b which yields the fastest asymptotic convergence. In practice, one cannot easily find this optimal b, so we also present heuristic rules for choosing b. Computational results appear quite promising.

BACKGROUND

Consider a finite state, discrete time, completely ergodic Markov process which is controlled by a decision maker. For each of the N states (i), at each transition of the process, the decision maker chooses an action $k = 1, \ldots, K_i$. This action results in transition probabilities p_{ij}^k, $j = 1, N$, and a reward (cost) q_i^k. p_{ij}^k is defined as the probability that the process, now in

state i and under policy k will move to state j over the next transition. q_i^k is defined as the expected reward (cost) over the next transition. The problem is to find the gain optimal action for each state.

Howard [5] showed that for a given policy set, the simultaneous set of linear equations,

$$(1) \qquad v_i + g = q_i^k + \sum_{j=1}^{N} p_{ij}^k v_j \qquad i = 1, \ldots, N$$

$$v_N = 0$$

could be solved to compute the gain g of the process. The v_i's are the relative rewards (costs) of starting the process in state i. Howard showed that the optimal gain could be obtained using a simple policy interative algorithm.

Consider a finite state, continuous time, completely ergodic Markov decision process. For each of the N states (i), at each transition, the decision maker chooses an action $k = 1, \ldots K_i$. This action results in a transition rate a_{ij}^k and a reward (cost) rate $\tilde{q}_i^k$. a_{ij}^k as defined as follows: In an increment of time dt, the process, now in state i and under policy k, will move to state j with probability $a_{ij}^k\, dt$ $(i \neq j)$. $\tilde{q}_i^k$ is the expected reward (cost) rate incurred over a residence in state i using action k.

Howard [5] showed that for a given policy set, the set of equations,

$$(2) \qquad g = \tilde{q}_i^k + \sum_{j=1}^{N} a_{ij}^k v_j, \qquad i = 1, \ldots, N,$$

$$v_N = 0$$

could be solved to compute the gain g and a policy iterative algorithm could be used to compute the optimal gain. Note that

$$a_{ii} = -\sum_{j \neq i} a_{ij}^k, \qquad i = 1, \ldots, N.$$

Finally, consider a finite state, completely ergodic semi-Markov decision process. The underlying Markov process has transition probabilities p_{ij}^k. The holding (transition) time (m) in going from state i to j is described by the density function $h_{ij}^k(m)$, $0 < m < \infty$. The expected holding time, given the system starts in state i is

$$T_i^k = \sum_{j=1}^{N} p_{ij}^k \int_0^\infty m h_{ij}^k(m)\, dm > 0$$

Jewell [6] showed that for a given policy set, the set of equations,

$$(3) \qquad v_i + T_i^k g = q_i^k + \sum_{j=1}^{N} p_{ij}^k v_j, \qquad i = 1, \ldots, N$$

$$v_N = 0$$

could be solved to compute the gain g and a policy iterative algorithm could be used to compute the optimal gain.

WHITE'S METHOD AND PROBLEM TRANSFORMATIONS

The bulk of the computational effort in policy iteration lies in solving (recursively) the set of equations (1), (2), (3). For large processes, techniques, such as Gaussian Elimination,

uickly become untenable. White [22] proposed a successive approximation approach for the ndiscounted, discrete time, Markov decision process[1]. Odoni [13] added bounds for g which e useful in termination decisions. The White-Odoni technique can be summarized as follows:

Assume we have computed sets of values $V_i(n-1)$, $v_i(n-1)$, $i = 1, \ldots, N$ and a uantity g_{n-1}. We then compute a new set

$$V_i(n) = \max_{1 \leqslant k \leqslant K_i} \left\{ q_i^k + \sum_{j=1}^{N} p_{ij}^k v_j (n-1) \right\},$$

$$g_n = V_M(n),$$

$$v_i(n) = V_i(n) - g_n,$$

$$L''(n) = \max_i \{V_i(n) - v_i(n-1)\}$$

$$L'(n) = \min_i \{V_i(n) - v_i(n-1)\}$$

here M is a state of the process such that for all sets of policies and some integer $u > 0$, the obability of reaching state M in u transitions, starting in any state i, is nonzero for all states i. hite showed that the repeated application of equations (4) will converge[1] to a solution for uations (1). Odoni showed that

$$L''(n) \geqq L''(n+1) \geqq g \geqq L'(n+1) \geqq L'(n).$$

In practice, White's algorithm has proven to be very effective for large scale systems. It is able, self-correcting, and, lends itself to the exploitation of any supersparsity [7].

While straight forward application of White's approach does not, in general, work for con-uous time, and semi-Markov processes, these processes can be transformed to a form com-tible with White's approach. Consider equations (2) with v_i added to both sides of the equa-n.

$$v_i + g = \tilde{q}_i^k + \sum_{j \neq i} a_{ij}^k v_j + (1 + a_{ii}^k) v_i, \qquad i = 1, \ldots, N,$$

$$v_N = 0$$

ting the definition of a_{ii}^k, then if

$$0 > a_{ii} = -\sum_{j \neq i} a_{ij}^k > -1, \qquad i = 1, \ldots, N,$$

uation (5) is of the same form as equation (1). Substituting $1 + a_{ii}^k$ for a_{ii}^k in the rate matrix, new matrix $\{a_{ij}^k\}$ has the properties of a stochastic matrix.

If (6) holds, it follows that White's method can be used to solve the continuous time rkov decision process. The following procedure can be used to convert a continuous time oblem to satisfy (6).

Let $a_{\max} = \max_{\substack{i=1,\ldots,N \\ k=1,\ldots,K_i}} |a_{ii}^k|$

[1] he assumptions used by White can be relaxed. Schweitzer [19] proved convergence for the general single chain acy-c process while Su and Deininger [20] extended this to the periodic case. Such conditions are hard to test in practice. R ently Platzman [14] has given a weaker condition that can be readily tested. Finally, Morton and Wecker [11] have g eralized most of the above plus have added some new dimensions to the algorithm.

2. Divide all a_{ij}^k and $\tilde{q}_i^k$, $i, j = 1, \ldots, N$, $k = 1, \ldots, K_i$ by $b > a_{\max}$. Condition (6) is now satisfied.

3. Using the new a_{ij}^k and $\tilde{q}_i^k$, solve the problem using White's method.

4. To express the results in terms of the original process, multiply the gain g by b. The optimal policy and relative rewards (costs), v_i, obtained are valid for the orginal process.

Note that the scaling really amounts to changing the time frame of the problem.

Consider the reorganization of equations (3).

$$g = \frac{q_i^k}{T_i^k} + \sum_{j \neq i} \frac{p_{ij}^k v_j}{T_i^k} + \frac{(p_{ii}^k - 1)}{T_i^k} v_i \qquad i = 1, \ldots, N, \tag{7}$$

$$v_N = 0$$

Letting

$$\tilde{q}_i^k = q_i^k / T_i^k,$$

$$a_{ij}^k = p_{ij}^k / T_i^k, \text{ and}$$

$$a_{ii}^k = (p_{ii}^k - 1) / T_i^k,$$

it is readily seen that equations (7) are of the same form as equations (2). As a consequence the transformation can also be applied to semi-Markov decision processes (the transformation is equivalent to Schweitzer's [17]). Note that a Markov process is itself a degenerate semi Markov process subject to transformation. One would consider such a transformation if the convergence properties of White's method could be improved. We now address this issue.

CONVERGENCE FACILITATION

There are several procedures that have been used in accelerating convergence in solving discounted Markov decision processes. By and, large, though, these have not been examined extensively in the non-discounted Markov decision process context. Briefly, the acceleration techniques include problem transformation, [17] cheap iterations [10, 23], suboptimal activity elimination [1, 2, 3, 8, 9, 15, 16, 21] and extrapolation procedures [23]. We will limit our discussion here to problem transformation.

In solving (generalized) discounted Markov decision processes, it is well known that the largest spectral radius of the transition matrices (i.e., the process spectral radius) governs the asymptotic convergence rate. Porteus [15], Totten [21] and others have devised problem transformations to reduce the process spectral radius. Morton and Wecker [11] have shown that asymptotic relative values and policy convergence are at least of order $(\alpha\lambda)^n$ where λ is greater than the subdominant eigenvalue[2] and $0 < \alpha < \infty$ is the discount factor. A reasonable question to ask is whether the choice of b in Step 2 can be made to reduce the modulus of the subdominant eigenvalue of the transition matrix of the optimal policy.

[2] The largest eigenvalue is always 1.0. The subdominant eigenvalue is the remaining eigenvalue having the largest modulus.

The transition matrix for policy δ resulting from the scaling procedure is

$$I + \frac{1}{b} A_\delta, \quad \text{where}$$

$$A_\delta = P_\delta - I.$$

et λ and $\bar{x}$ be an eigenvalue and associated eigenvector, respectively, of the starting transition atrix $I + \frac{1}{a_{\max}} A_\delta$. Then

)
$$\frac{a_{\max}}{b} \lambda + \frac{b - a_{\max}}{b}$$

an eigenvalue of $I + \frac{1}{b} A_\delta$ with $\bar{x}$ its associated eigenvector. Now clearly

$$\frac{a_{\max}}{b} \operatorname{re}\lambda + \frac{b - a_{\max}}{b} \geqq \operatorname{re}\lambda$$

here $\operatorname{re}\lambda$ is the real part of λ with $-1 \leqq \operatorname{re}\lambda \leqq 1$ and $b > a_{\max} \geqq 0$. However, it may not be ue that

)
$$\left| \frac{a_{\max}}{b} \lambda + \frac{b - a_{\max}}{b} \right| \geqq |\lambda|.$$

ppose δ indexes an optimal policy and λ is a subdominant eigenvalue associated with this pol-. Expanding the square of the modulus of both sides of (9) with $\lambda = \lambda_1 + \lambda_2 i$ gives that a duction in the modulus of λ requires

$$(1 - \lambda_1) \left[\lambda_1 + \frac{b - a_{\max}}{a_{\max} + b} \right] \leqq \lambda_2^2.$$

$\lambda_2 = 0$, then either $\lambda_1 = 1$ and no reduction can be made or $\lambda_1 \leqq (a_{\max} - b)/(a_{\max} + b)$ and is necessarily negative. In this case, it would appear that any $b > a_{\max}$ will yield a resultant nefit in asymptotic convergence. However, this is not necessarily true, since we may "bump" o another eigenvalue. That is, increasing b to decrease the absolute value of the dominant egative) eigenvalue will eventually result in some other (positive) eigenvalue increasing until ecomes the new subdominant eigenvalue. At that point further increases in b will not im-ve the convergence rate.

As an example, consider the Markov process whose transition matrix is given as follows:

$$\begin{bmatrix} .31 & .13 & .21 & .05 & .10 & .20 \\ .15 & .12 & .16 & .20 & .12 & .25 \\ .02 & .01 & .01 & .01 & .93 & .02 \\ .12 & .28 & .09 & .16 & .04 & .31 \\ 0 & .01 & .85 & 0 & .09 & .05 \\ .11 & .30 & .10 & .15 & .14 & .20 \end{bmatrix}$$

The eigenvalues are 1.0, −.8421, .6945, .2079, −.085 + .0116i, and −.085 −.0116i. It uld appear that problem transformation should be of value in speeding convergence, since l subdominant eigenvalue is negative. From the preceding development, it would be ected that the convergence rate of the process would be maximized at the value of "b" which ults in the largest negative eigenvalue being equal to the largest positive eigenvalue. Apply-i equation (8), to equate the two eigenvalues of the transformed matrix, we get

$$\frac{.99}{b}(.8421) - \frac{b-.99}{b} = \frac{.99}{b}(.6945) + \frac{b-.99}{b}$$

Solving, we get $b = 1.063$. In other words, transforming the process using $b = 1.063$ should achieve the "best" asymptotic convergence for the process. As a test, White's Algorithm was run using costs of

$$\bar{q} = (1.14,\ 2.27,\ 5.06,\ 2.97,\ 3.96,\ 4.90)$$

(only one policy per state). The problem was declared "solved" when $L''(n) - L'(n) \leqslant 10^{-4}$. Runs were made for various values of b (see Figure 1). The actual minimum number of iterations (30) occurred for a value of $b \approx 1.09$, whereas the number of iterations for $b \approx 1.063$ was slightly higher (31). The inaccuracy in prediction is expected, since the method of prediction considers only main effects and ignores the contribution of the smaller eigenvalues.

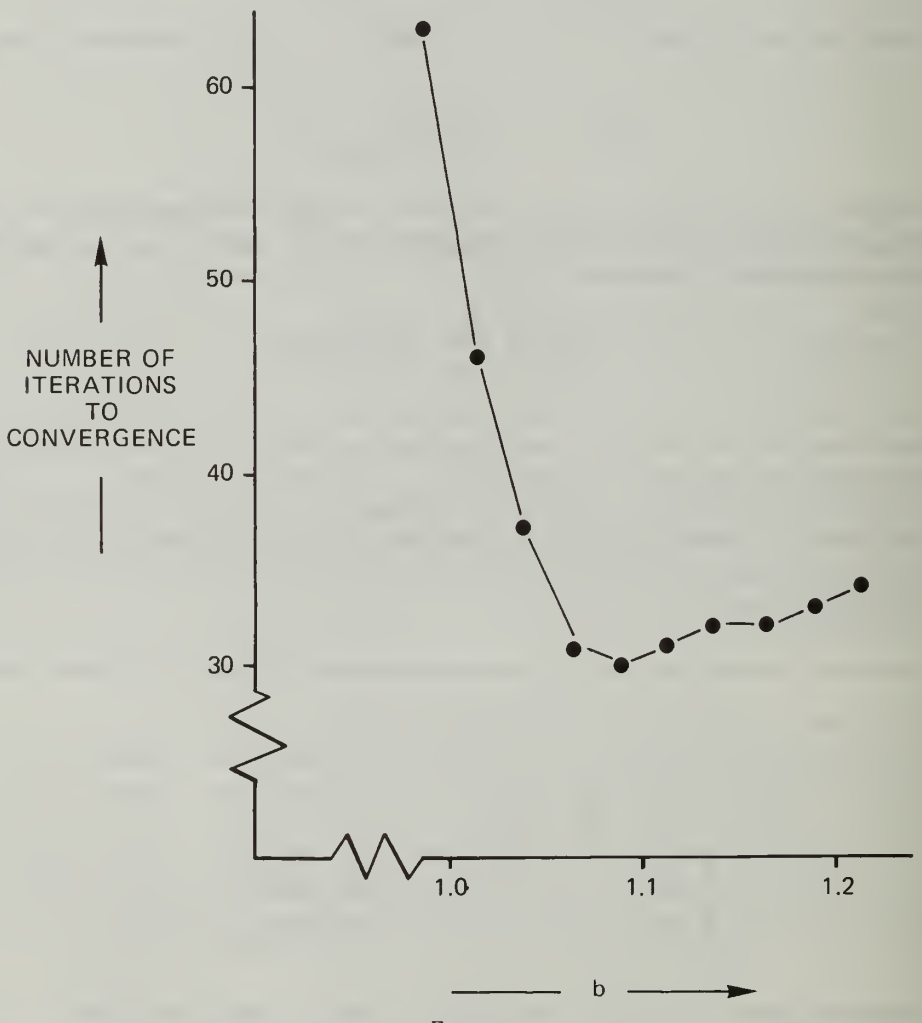


FIGURE 1.

As one might expect, the straightforward application of the above observations is not practical, since the determination of eigenvalues for large processes is itself difficult. However, in practice it is usually intuitively obvious to the analyst that a process may possess strong cyclic tendencies, indicating that some eigenvalue has a large negative real component. If the cyclic tendency is strong enough, this eigenvalue will be the subdominant eigenvalue and the above

evelopment suggests that some $b > a_{max}$ may decrease the resulting asymptotic convergence ıte. In any event, applying White's method, using several values of b marginally larger than max, and noting the convergence rate of the process for various values of b can many times be f value.

In testing the above we noted that if b was made successively slightly larger than a_{max}, ıther the convergence improved dramatically or the convergence slightly deteriorated. To ırther test this observation, we randomly generated Markov decision problems with the umber of states varying from 3 to 20. Within each state ten different actions were available. /hite's method was used to solve each using b values of

$$b_0 = a_{max} + 10^{-5}$$

$$b_1 = 1.05b_0$$

$$b_2 = 1.10b_0$$

$$b_3 = 1.15b_0$$

gain, problems were declared "solved" at iteration n when $L''(n) - L'(n) \leqq 10^{-4}$. If a prob- :m was solved in fewer iterations for some b_i than b_j with $i > j$, then the problem transforma- on was declared beneficial. Otherwise the transformation was classified as non-beneficial. learly a problem could be mislabeled as non-beneficial using the grid given above but may in ıct be beneficial for some $b > a_{max}$. The opposite is not the case.

Table 1 gives the total number of iterations to solve the non-beneficial and beneficial roblem cases. N_n and N_b stand for the number of problems labelled non-beneficial and eneficial, respectively. If we can assume that the average performance of the set of randomly enerated problems used in this study is representative of the performance of the set of real orld problems, then the following observations can be made. First, problems whose conver- ence can be improved by increases in b and above a_{max} are those problems that are hard to olve anyway (see Table 1, 19.5 versus 35.4 iterations). Second, when a problem does not ıow convergence improvement when b is increased above a_{max}, the deterioration in conver- ence speed is not dramatic (see Table 1, 19.5 versus 22.8 iterations for a 15% increase in b ove a_{max}). Finally, convergence improvements, when they occur, are rather dramatic (see able 1, 35.4 to 18.5 iterations for a 15% change in b above a_{max}). These observations suggest ıat use of problem transformation can be of significant value in speeding convergence.

TABLE 1. *Summary of Iteration Counts*

N_n / N_b	Total Iteration Counts			
	b_0	b_1	b_2	b_3
46	896	942	997	1047
67	2372	1520	1310	1241
Average per Problem	19.5	20.5	21.7	22.8
	35.4	22.7	19.6	18.5

BIBLIOGRAPHY

[1] Hastings, N.A.J., "A Test for Non-Optimal Actions in Undiscounted Finite Markov Decision Chains," Management Science, *23*, No. 1, pp. 87-92 (1976).

[2] Hastings, N.A.J. and J.M.C. Mello, "Erratum Tests for Suboptimal Actions in Discounted Markov Programming," Management Science, *20*, No. 17, p. 1143 (1974).
[3] Hastings, N.A.J. and J.M.C. Mello, "Tests for Suboptimal Actions in Discounted Markov Programming," Management Science, *19*, No. 9, pp. 1019-1022 (1973).
[4] Hordijk, A. and H. Tijms, "The Method of Successive Approximations and Markovian Decision Problems," Operations Research, *22*, pp. 519-521 (1974).
[5] Howard, R.A., *Dynamic Programming and Markov Processes* (MIT Press and Wiley, New York, 1960).
[6] Jewell, W.S., "Markov Renewal Programming: I and II," Operations Research Society of America, *11*, pp. 938-971 (Nov.-Dec., 1963).
[7] Kalan, J.E., "Aspects of Large-Scale, In-Core Linear Programming," Proceedings of ACM Annual Conference, Chicago, Illinois (August 3-5, 1971).
[8] MacQueen, J., "A Modified Dynamic Programming Method for Markovian Decision Problems," Journal of Mathematical Analysis and Applications, *14*, pp. 38-43 (1966).
[9] MacQueen, J.B., "A Test for Suboptimal Actions in Markovian Decision Problems," Operations Research, *15*, pp. 559-561 (1967).
[10] Morton, T.E., "On the Asymptotic Convergence Rate of Cost Differences for Markovian Decision Processes," Operations Research, *19*, pp. 244-248 (1971).
[11] Morton, T.E. and W.E. Wecker, "Discounting, Ergodicity, and Convergence for Markov Decision Processes," Management Science, *23*, pp. 890-900 (1977).
[12] Nering, E.D., *Linear Algebra and Matrix Theory*, (2nd. Ed., John Wiley and Sons, New York, 1970).
[13] Odoni, A.R., "On Finding the Maximal Gain for Markov Decision Processes," Operations Research, *17*, pp. 857-860 (1969).
[14] Platzman, L., "Improved Conditions for Convergence in Undiscounted Markov Renewal Programming," Operations Research, *25*, No. 3, pp. 529-533 (1977).
[15] Porteus, E.L., "Bounds and Transformations for Discounted Finite Markov Decision Chains," Operations Research, *23*, No. 4, pp. 761-784 (1975).
[16] Porteus, E.L., "Some Bounds for Discounted Sequential Decision Processes," Management Science, *18*, No. 1, pp. 7-11 (1971).
[17] Schweitzer, P.J., "Iterative Solution of the Functional Equations of Undiscounted Markov Renewal Programming," Journal of Mathematical Analysis and Applications, *34*, pp. 495-501 (1971).
[18] Schweitzer, P.J., "Multiple Policy Improvements in Undiscounted Markov Renewal Programs," Operations Research Society of America, *19*, pp. 784-793 (May-June, 1971).
[19] Schweitzer, P.J., "Perturbation Theory and Markovian Decision Processes," MIT Operations Research Technical Report, *15*, (June 1965).
[20] Su, S.Y. and R.A. Deininger, "Generalization of White's Method of Successive Approximations to Periodic Markovian Decision Processes," Operations Research, *20*, No. 2, pp. 318-326 (1972).
[21] Totten, J.C., "Computational Methods for Finite State Finite Valued Markovian Decision Problems," Operations Research Center, University of California, Berkeley, ORC-71 (1971).
[22] White, D.J., "Dynamic Programming, Markov Chains, and the Method of Successive Approximations," Journal of Mathematical Analysis and Applications, *6*, pp. 373-376 (1963).
[23] Zaldivar, M. and T.J. Hodgson, "Rapid Convergence Techniques for Markov Decision Processes," Decision Sciences, *6*, pp. 14-24 (1975).

AN ALGORITHM (GIPC2) FOR SOLVING INTEGER PROGRAMMING PROBLEMS WITH SEPARABLE NONLINEAR OBJECTIVE FUNCTIONS

Claude Dennis Pegden

The Pennsylvania State University
University Park, Pennsylvania

Clifford C. Petersen

Purdue University
W. Lafayette, Indiana

ABSTRACT

This paper presents an algorithm for solving the integer programming problem possessing a separable nonlinear objective function subject to linear constraints. The method is based on a generalization of the Balas implicit enumeration scheme. Computational experience is given for a set of seventeen linear and seventeen nonlinear test problems. The results indicate that the algorithm can solve the nonlinear integer programming problem in roughly the equivalent time required to solve the linear integer programming problem of similar size with existing algorithms. Although the algorithm is specifically designed to solve the nonlinear problem, the results indicate that the algorithm compares favorably with the Branch and Bound algorithm in the solution of linear integer programming problems.

. INTRODUCTION

This paper presents an algorithm for solving the following nonlinear pure integer programming problem.

$$\text{Max } g(x) = \sum_{j=1}^{NNS} f_j(x_j) + \sum_{j=NNS+1}^{NS} c_j x_j \qquad (1)$$

$$\text{S.t. } Ax \lessgtr b$$
$$x \in I^+$$

where: c, A, and b denote the usual constant arrays

I^+ denotes the set of all nonnegative integers

$\lessgtr$ denotes constraints of the less-than-or equal type and greater-than-or-equal type

$f_j(x_j)$ is a single variable nonlinear function with $f_j(0) = 0$

The region defined by $Ax \lessgtr b$, $x \in I^+$ is bounded and nonempty

NNS denotes the number of nonlinear stages

NS denotes the total number of stages

There are several transformations which are useful to convert problems to the required orm. If the problem contains k equality constraints of the form $a_i x = b_i$, we can replace this et by a set of $k + 1$ inequalities of the form $a_i x \leqslant b_i$ for $i = 1, \ldots, k$ and

$\left(\sum_{i=1}^{k} a_i\right) x \geqslant \sum_{i=1}^{k} b_i$. If the problem contains one or more nonlinear functions $f_j(x_j)$ such that $f_j(0) \neq 0$, we can replace each by a new function $f_j'(x_j) = f_j(x_j) - f_j(0)$. If the nonlinear portion of the objective function cannot be separated into functions of a single variable, but the nonseparable portion can be separated into k functions of linear integer combinations of the variables, we can convert the problem to the required form by replacing each of the k linear combinations in the objective function by a dummy variable d_k. The dummy variables are forced to assume the appropriate values by appending, for each k, the constraint that d_k equals the k^{th} linear combination. To illustrate, consider the following example:

$$\begin{aligned} &\text{Max } (x_1 + 3x_2)^2 - 9x_2^2 \\ &\text{S.t. } 2x_1 + x_2 \leqslant 5 \\ &x_1 \text{ and } x_2 \in I^+ \end{aligned} \tag{2}$$

To convert the problem to the desired form we must express the objective function as the sum of nonlinear functions of a single variable. We accomplish this by replacing the linear combination $x_1 + 3x_2$ in the objective function by the dummy variable d_1 and append the constraint that $d_1 = x_1 + 3x_2$, yielding the following equivalent problem:

$$\begin{aligned} &\text{Max } d_1^2 - 9x_2^2 \\ &\text{S.t. } 2x_1 + x_2 \leqslant 5 \\ &d_1 - x_1 - 3x_2 = 0 \\ &d_1,\ x_1,\ x_2 \in I^+ \end{aligned} \tag{3}$$

If the objective function contains a product term of two variables, we can employ the device of completing the square to transform the problem to the desired form. To illustrate, consider the following nonlinear integer programming problem:

$$\begin{aligned} &\text{Max } x_1^2 + 6x_1x_2 \\ &\text{S.t. } 2x_1 + x_2 \leqslant 5 \\ &x_1,\ x_2 \in I^+ \end{aligned} \tag{4}$$

At first glance, because of the product term $6x_1x_2$, the objective function appears to be nonseparable. However, by adding and subtracting $9x_2^2$ to the objective function we complete the square, and by factoring the objective function becomes:

$$(x_1 + 3x_2)^2 - 9x_2^2$$

The problem is now identical to the previous example and is convertible to the desired form by the introduction of a dummy variable.

The problem given by (1) is very difficult to solve with existing methods. If the problem contains only one or two constraints, it may be amenable to solution by dynamic programming. If all f_j are nondecreasing functions and c and A are nonnegative, then the imbedded state space approach presented by Morin and Marsten [7,8] may be employed to help mitigate the "curse of dimensionality" normally encountered in problems having several constraints. Also, if the problem is of very small size and can be converted (using the binary expansion) to a zero-one polynomial problem [11], a solution may be obtainable using either the transformation of Watters [13], or a zero-one polynomial algorithm such as that given by Taha [12]. However, many nonlinear integer programming problems of both practical and theoretical significance fall

ıto neither class and are therefore essentially unsolvable by methods other than the GIPC2 lgorithm presented here.

. OVERVIEW OF THE ALGORITHM

The GIPC2 algorithm is based upon the notion that although the solution space of an ıteger programming problem may be large, it is finite. The general approach of the algorithm ; to implicitly enumerate, by means of a fathoming test, a set of candidate solutions to the roblem. The set of candidate solutions is defined in such a way that it necessarily contains the ptimal solution to the problem. The general phases of the algorithm are as follows:

I. Find a good feasible solution to the problem.

II. Determine a vector of upper and lower bounds on x.

III. Generate a set of candidate solutions to the problem. This set should be as small as possible, while necessarily containing the optimal solution.

IV. Implicitly search the set of candidate solutions for the optimal solution to the problem.

Note that in developing an implicit enumeration algorithm for the zero-one integer programming problem, the special structure of the problem can be exploited to eliminate Phases II nd III. The set of candidate solutions can be simply defined as the set of vectors produced by ll combinations of assignments of zero and one to each variable in the problem. The major ısk of the Balas algorithm [1] consists of essentially Phase IV; implicit enumeration of candiate solutions. However in the nonlinear integer programming problem given by (1), our task ; more difficult. If we define the set of candidate solutions as simply all combinations of feasile integer values assigned to each variable, the number of candidate solutions can become so ırge, for some problems, as to make the approach computationally intractable. The key to the uccess of the GIPC2 algorithm, therefore, is the ability of the procedure to limit the set of andidate solutions to a manageable size, while guaranteeing that the optimal solutions is conıined within the set.

The steps of the algorithm require the solution of several linear programming problems or obtaining bounds on the optimum nonlinear solution. Our approach to solving the nonnear problem consists essentially of substituting one of three different linear approximating unctions for the nonlinear objective function at each step in the algorithm where a linear proramming solution is required. The three linear approximating functions are defined as follows:

$c_a x$: a "good" linear approximating function to the nonlinear objective function. The linear function $c_a x$ does not necessarily bound the nonlinear function above or below.

$c_l x + \alpha_l$: a "good" lower bounding linear approximating function to the nonlinear objective function. For all x in the domain,

$$c_l x + \alpha_l \leqslant g(x)$$

$c_u x + \alpha_u$: a "good" upper bounding linear approximating function to the nonlinear objective function. For all x in the domain,

$$c_u x + \alpha_u \geqslant g(x)$$

All the linear programming solutions in the algorithm are used exclusively to obtain either a feasible solution or an upper or lower bound on the optimal solution. By appropriately selecting our linear approximating function for each linear programming problem we set bounds that narrow the range of search. Since the feasible region is not altered, the algorithm guarantees an exact solution to the nonlinear integer programming problem.

A number of excellent methods exist for computing a linear approximation to a separable nonlinear function. These include a least squares fit procedure and a linear programming formulation to minimize either the sum of the absolute values or the maximum deviation. Geoffrion [3] discusses the use of objective function approximations in mathematical programming and presents methods for determining the "best" approximation for cases of particular interest in mathematical programming. GIPC2 employs a less elegant approximating procedure, but because of its simplicity and the nature of the problem, the procedure is well suited for this particular application. It should be noted that although the specific approximation employed will not affect the accuracy of the final solution obtained by the GIPC2 algorithm, poor approximations will have the consequence of increasing the computation time and storage requirements of the algorithm.

3. FINDING A GOOD FEASIBLE SOLUTION (PHASE I)

Our objective is to compute SMIN, a lower bound on the optimal objective function value, by finding a good feasible solution to the problem. We accomplish that by the following steps:

1. Replace $g(x)$ by $c_a x$ and solve the resulting linear programming problem.
2. Force the continuous solution to a good feasible integer point, x, by successively testing each fractional variable at its rounded down and rounded up value, then fixing the variable at the integer point associated with the largest feasible value of the objective function.
3. Compute SMIN by substituting x^0 into the nonlinear objective function. If step 2 fails to yield a feasible integer solution, SMIN is set to a large negative number or may be specified on data input if a lower bound is known.

4. COMPUTING UPPER AND LOWER VARIABLE BOUNDS (PHASE II)

Once SMIN has been obtained, we next establish good upper and lower bounds on the variables so that the range of search may be narrowed. We do this by solving two linear programming problems of the form maximize x_j and minimize x_j, subject to the original constraints of the problem and the additional constraint that the objective function be greater than or equal to SMIN. For the nonlinear objective, this procedure would produce a nonlinear constraint which we desire to avoid. By noting that since $c_u x + \alpha_u \geqslant g(x)$, then $c_u x + \alpha_u \geqslant \text{SMIN}$, we will replace the nonlinear constraint $g(x) \geqslant \text{SMIN}$ with a series of linear constraints which conservatively approximate the single nonlinear constraint. The procedure for computing upper and lower bounds on x with a nonlinear objective function is as follows for each variable in the problem:

1. Determine initial variable bounds by solving the following two linear programming problems:

$$\begin{array}{lll} \text{Max} & x_j & \\ \text{S.t.} & Ax \leqslant b & \\ & c_u x \geqslant \text{SMIN} - \alpha_u & \end{array} \qquad \begin{array}{lll} \text{Min} & x_j & \\ \text{S.t.} & Ax \leqslant b & \\ & c_u x \geqslant \text{SMIN} - \alpha_u & \end{array}$$

2. Compute $c_u x + \alpha_u$ based on the current variable bounds.

3. Using the new $c_u x + \alpha_u$, generate a new constraint $c_u x \geqslant \text{SMIN} - \alpha_u$ and append it to both linear programming problems.

4. Solve the two linear programming problems to determine new variable bounds. If the variable bounds have been improved as shown by a reduction in domain, a stronger "cut" may be possible, so go to step 2.
Otherwise terminate the procedure and use the current variable bounds, UB_j and LB_j.

The procedure will obviously terminate at some point with no improvement; possibly with bounds that uniquely determine the value of some of all variables. Although the number of iterations required is problem dependent, the procedure typically converges within two or three iterations.

5. GENERATING THE SET OF CANDIDATE SOLUTIONS (PHASE III)

In Phase III we enumerate in an efficient manner solutions that yield a value equal to or greater than SMIN, possibly including some solutions that are infeasible. In Phase IV we will identify the optimal (feasible) solution.

It is convenient to transform each domain LB_j to UB_j found in Phase II into a domain 0 to $(UB_j - LB_j)$. This is done by defining a new vector y as $y = x - LB$ and replacing x in our original problem with $LB + y$. Also, if the lower bound and upper bound are equal for any variable j, we can delete y_j from the problem as we know its optimal value is zero (the optimal value of $x_j = LB_j = UB_j$). Our problem now becomes:

$$\begin{aligned} &\text{Max } G(y) + g(LB) \\ &\text{S.t. } Ay \lessgtr \bar{b} \\ &y \in I^+ \end{aligned} \tag{5}$$

where $G(y) = g(y + LB) - g(LB)$ where the bar over the constant array b denotes modification of the original values due to the substitution $y = x - LB$. From Phases I and II we also know:

$$\begin{aligned} &G(y) \geqslant \overline{\text{SMIN}} \\ &0 \leqslant y \leqslant \overline{UB} \end{aligned} \tag{6}$$

where

$\overline{\text{SMIN}}$ the new lower bound on the objective function after the transformation from x to y is $\overline{\text{SMIN}} = \text{SMIN} - g(LB)$

$\overline{UB}$ is the vector of upper bounds on y and is equal to $UB - LB$.

Our procedure for generating the set of candidate solutions to problem (5) consists of enumerating all y vectors satisfying the conditions given by (6) and the constraint set $Ay \lessgtr \bar{b}$. Note that the optimal y vector will satisfy all the above conditions and therefore will necessarily be contained within the set of candidate solutions. In order to facilitate the computations, we relax the condition $Ay \lessgtr \bar{b}$ at several points in the procedure. As a result of this relaxation,

the set of candidate solutions which is generated may contain entries which are not feasible to our problem. This relaxation allows us to accomplish the enumeration of y vectors in a recursive tabular fashion akin to the procedure employed in discrete dynamic programming. However, Bellman's "Principle of Optimality" is never invoked in the process and, therefore, the assumption of monotonicity is not required in the development. Because of the similarities between discrete dynamic programming and the recursive tabular procedure employed here for enumerating the y-vectors, it will aid our discussion to borrow the following dynamic programming terminology.

STAGE:	a function of a single variable
STATE:	the state at stage k is the value of $G(y)$ resulting from an assignment of integer values to y_k at stage k through y_n at the last stage, inclusive
DECISION:	a positive integer assignment to an element of y
NDEC:	the number of decisions made at a given stage-stage

Our general procedure is to generate a table for each stage containing all potentially optimal states and the corresponding decisions at that stage which, in conjunction with prior decisions, produce that state value. By the means of certain tests, we exclude a large number of entries from the tables by ascertaining that they are either infeasible or nonoptimal to our problem. Assignments which the tests fail to exclude, and are thus contained in the tables are termed as candidate solutions to our problem.

The computations begin at the last stage (n) and recursively proceed to stage 1. The stage n computations are performed as a special case, with the computations for stages $n-1$, $n-2$, ----k, ----, 1 proceeding as the general case. Therefore details of the stage n and stage k computations are sufficient to fully describe the algorithmic procedure for generating the set of candidate solutions to the problem.

Stage n Computations

In the stage n computations we simply enumerate in tabular form all possible state values for the integer domain of y_n. The following table is produced.

STAGE n

STATE	NDEC	DECISIONS
0	1	0
$G(1)$	1	1
$G(2)$	1	2
·	1	3
·	·	·
·	·	·
·	·	·
·	·	·
$G(\overline{UB}_n)$	1	$\overline{UB}_n$

Stage k Computations

The general stage computations begin by forming and initializing two vectors, named TVEC and LVEC. The first records the total state value for each possible decision, and the second records location information relative to the previously generated stage. These vectors are used simply to produce efficiently the STATE, NDEC, DECISIONS table for each state.

The vectors TVEC and LVEC are initially dimensioned equal to the number of possible decisions at stage k. The d^{th} entry in TVEC and LVEC corresponds to the decision $y_k = d$, with d initially ranging from 0 to $\overline{UB}_k$. However, we will show that as the state value at a given stage increases, the number of possible decisions at that stage decreases. We will take advantage of this property to continuously reduce the dimension of TVEC and LVEC as the computations for stage k proceed.

Each entry in TVEC, corresponding to a given decision d assigned to y_k, is the total state value at stage k. The total state value for each decision is comprised of a fixed state contribution at stage k combined with the i^{th} total state at stage $k+1$, where i is given in the corresponding location in LVEC. Defining $S_{k+1,i}$ as the i^{th} state value in the stage $k+1$ table, all possible total state values at stage k resulting from $y_k = d$ are given by:

$$t_d(i,\ k) = g_k(d + LB_k) - g_k(LB_k) + S_{k+1,i} \tag{7}$$

where i is defined from 1 to the number of state values in the stage $k+1$ table and g_k denotes the objective function for the k^{th} stage. Note that by systematically indexing (7) over all i for each entry in TVEC we can generate in ascending order of magnitude all possible state values and the corresponding decisions for stage k. This recursive relationship, in conjunction with two exclusion tests, is the basis for generating the set of candidate solutions to the problem. The purpose of the two exclusion tests is to exclude as many states and corresponding decisions as possible from the stage k table by discerning that they are either infeasible or nonoptimal to the problem.

Exclusion Test A

At each stage k, solve the following LP.

$$\begin{aligned} V = \min \quad & \sum_{j=k}^{n} c_{lj} y_j + \alpha_{lj} \\ S.t. \quad & c_\mu y \geqslant \overline{\text{SMIN}} - \alpha_\mu \\ & Ay \lesseqgtr \bar{b} \\ & y \geqslant 0 \end{aligned} \tag{8}$$

where c_l, c_μ, and α_l and α_μ are the constants from the previously defined linear bounding functions and the subscript j is used to denote the j^{th} stage. Exclude all state values for which $t_l(i,\ k) < V$, and revise the lower bound $\overline{LB}_k$, accordingly. The optimal integer solution Y^* of the n-variable problem will be a point within the feasible region given in (8). It will have a state value at stage k, as given by the objective of (8), when only its Y_j^* for $j = k,\ k+1,\ \ldots,\ n$ are considered. V is a lower bound on the minimum state value at stage k considering all of the points in the feasible region. It follows that y_j^*, $j = k,\ k+1,\ \ldots,\ n$, the optimal decisions at stages k through n, will yield a state value $\geqslant V$ and will not be excluded by discard of all state values $< V$.

Exclusion Test B

At each stage k, solve the following two LP's, one with y_k fixed at its current lower bound and the other with y_k fixed at its current upper bound.

$$
\begin{array}{lll}
 & W = \max \sum_{j=k}^{n} c_{\mu j} y_j + \alpha_{\mu j} & Z = \max \sum_{j=k}^{n} c_{\mu j} y_j + \alpha_{\mu j} \\
 & S.t. \quad c_\mu y \geqslant \overline{\text{SMIN}} - \alpha_\mu & S.t. \quad c_\mu y \geqslant \overline{\text{SMIN}} - \alpha_\mu \\
(9)\ (10) & \quad A y \lesseqgtr \bar{b} & \quad A y \lesseqgtr \bar{b} \\
 & \quad y_k = \overline{LB}_k & \quad y_k = \overline{UB}_k
\end{array}
$$

where: $\overline{LB}_k$ denotes the current lower bound on y_k (initially 0)
$\overline{UB}_k$ denotes the current upper bound on y_k (initially $UB_k - LB_k$).

Exclusion test B is dynamic in the sense that the bounds on y_k are continuously tightened as larger and larger states values are generated by the algorithm. This tightening of bounds is accomplished as follows:

(a) if the current state $> W$ or if the problem is infeasible replace $\overline{LB}_k$ by $\overline{LB}_k + 1$ and compute the new value of W

(b) if the current state $> Z$ or if the problem is infeasible replace $\overline{UB}_k$ by $\overline{UB}_k - 1$ and compute the new value of Z

Recall the general process of generating candidate solutions using the TVEC and LVEC vectors. At stage k candidate values (d) are assigned to y_k starting with the current LB_k. A state value $\sum_{j=k}^{n} G_j(y_j)$ will result. However, if it exceeds W, an upper bound on the maximum feasible state value with y_k equal to $\overline{LB}_k$, or if there is no feasible solution to problem (9), then $\overline{LB}_k$ is clearly not a valid assignment. The lower bound may be increased by one, to seek a feasible solution and/or a new increased value of W. Based on similar reasoning, the current upper bound $\overline{UB}_k$ may be tightened, by reducing it by one, whenever the state values exceeds Z, an upper bound on the maximum feasible state value with y_k equal to $\overline{UB}_k$, or whenever there is no feasible solution to problem (10). The proof that the optimal state value and corresponding decision y_k^* will not be excluded follows from the fact that initially $\overline{LB}_k \leqslant y_k^* \leqslant \overline{UB}_k$. As larger and larger state values are generated, the bounds on y_k will tighten until the upper and lower bounds are equal to y_k^*. The bounds cannot be tightened to exclude y_k^* since y_k^* is feasible to the constraint set given in (9) and (10) therefore the corresponding state must be less than or equal to W and Z.

It should be noted that although we must recompute the values of W or Z when we tighten the upper or lower bound on y_k, there is no need to solve the entire LP given in (9) or (10) again. Since we are only changing one of the right hand side constants of the original LP, we can make use of the basis inverse to update the final tableau and employ the dual simplex algorithm when necessary to regain feasibility.

The step-by-step procedure for generating the stage k table is as follows:

1. Compute the lower bound V as given by (8).

2. Form and initialize the vectors TVEC and LVEC where the d^{th} entry in TVEC is given by

$$t_d = g_k(d + LB_k) - g_k(LB_k) + S_{k+1,i}$$

where d initially ranges from 0 to $\overline{UB}_k$ and where i is chosen such that t_d is the minimum value greater than or equal to V. (Exclusion Test A.) The value of i is recorded as the d^{th} entry in LVEC.

3. Compute upper bounds W and Z as defined by (9) and (10) and use them to eliminate any infeasible decisions.

4. Flag all entries having the smallest state value in TVEC.

5. If the smallest state value is less than or equal to both W and Z, go to step 6. Otherwise, apply Exclusion Test B to tighten bounds on y_k. If $\overline{LB}_k > \overline{UB}_k$ the stage k table is complete. Otherwise go to step 4.

6. Enter values for the STATE, NDEC, and DECISIONS as one row of the stage k table.

7. Update the flagged entries in the TVEC to the next largest possible state value for that decision by increasing i by 1 and update LVEC accordingly. If the d^{th} entry in TVEC is flagged, the updated TVEC (t_d) and LVEC (i_d) are given by

$$t_d = t_d + S_{k+1,i+1} - S_{k+1,i}$$

$$i_d = i_d + 1$$

Go to step 4.

Example

To illustrate the computations in generating the set of candidate solutions, consider the following problem:

Enumerate: $y_1 + 3y_2 + 2y_3 \geqslant 20$
$0 \leqslant y_1 \leqslant 5$
$0 \leqslant y_2 \leqslant 6$
$0 \leqslant y_3 \leqslant 6$
$y \in I^+$

where the constraint set $Ay \lessgtr \bar{b}$ is:
$y_1 + y_2 + y_3 \leqslant 8$

The computations follow the step-by-step procedure outlined above (beginning at stage 3) and produce the following tables:

STAGE 3			STAGE 2			STAGE 1		
STATE	NDEC	DECISIONS	STATE	NDEC	DECISIONS	STATE	NDEC	DECISIONS
0	1	0	18	2	4, 6	20	3	0, 1, 2
2	1	1	19	1	5	21	2	0, 1
4	1	2	20	2	4, 6	22	2	0, 1
6	1	3	21	1	5			
8	1	4	22	1	6			
10	1	5						
12	1	6						

Candidate solutions are recovered from the tables by tracking through the tables beginning at Stage 1 and working towards the last stage. This tracking process can be thought of as generating a combinatorial "tree" of solutions for a specified starting or "goal" state. The nodes of the tree correspond to a given stage and state, and the branches emanating from the node correspond to the alternate decisions for that stage and state. A path through the tree represents an assignment of integer values to each stage of the problem. For example, with a state value of 22 we have two candidate solutions $y_1 = 0$, $y_2 = 6$, $y_3 = 2$ and $y_1 = 1$, $y_2 = 6$, $y_3 = 3$, the latter being non-feasible to the $Ay \leqslant b$ constraint.

6. IMPLICIT ENUMERATION OF CANDIDATE SOLUTIONS (PHASE IV)

In generating the set of candidate solutions, we have excluded only state values and assignments which could be shown to be infeasible or non-optimal to our problem. Therefore the optimal feasible solution to our problem is necessarily contained within the set of feasible and possibly infeasible solutions. Our strategy is to search for a solution within the set which is feasible with respect to the constraints of our problem. To guarantee that the first feasible solution found is also optimal, the search is performed starting with the largest state value at Stage 1 and working towards the smallest state value at Stage 1.

For a given goal state, the number of candidate solutions is simply the number of paths emanating from the corresponding state of Stage 1. For simple trees explicit enumeration, by substituting each candidate solution into the constraint set of the problem and testing for feasibility, is quite practical. Due to the combinatorial nature of the tree, this approach can become computationally overburdening for larger problems. We will therefore employ a method for implicity evaluating candidate solutions. Through the application of a fathoming test, large portions of the combinatorial tree will be exempted from enumeration.

The implicit evaluation procedure starts by selecting the largest state value at Stage 1; this is termed the present goal and there will be a corresponding tree. The examination of paths through the tree is performed by the systematic assignment of values to the y-variables at each stage, starting at the first stage by assigning a value to y_1. A partial evaluation at Stage j is defined as the assignment of integer values from the first stage node through the j^{th} stage node, inclusive. The state contribution resulting from this partial evaluation is designated ZINT. All paths through the tree will yield the present goal, but not all paths will yield y-values that satisfy the constraints of the original problem.

Our purpose is to devise a test to detect as early as possible in the search if a particular branch (and its sub-branches) cannot yield a candidate solution feasible to the constraints of the original problem. We accomplish this by comparing the goal state value to the sum of ZINT and the continuous maximal solution (ZCONT) for the y-variables not yet assigned values. Since ZCONT is greater than or equal to any feasible integer completion for the unassigned y-variables, if the sum of ZINT plus ZCONT is less than the present goal state, we can exclude all integer completions of this partial evaluation from consideration. At this point the branch is said to be "fathomed" and we "backtrack", that is, go back to the preceding node and evaluate the remaining branches emanating from it.

If the present goal is achieved, by completing a path through the tree without violating any constraints of the original problem, the current assignment to y is added to the lower bound vector (LB) of x and the algorithm terminates. If the present goal is not achieved, that is, if no feasible path through the tree exists, then the next largest state value at Stage 1 is used as the goal state and its applicable tree is searched. The algorithm will terminate because the range of state values generated is bounded such that it included at least one feasible y-vector.

The implicit evaluation method described above requires a computationally efficient procedure for computing ZINT + ZCONT at each partial evaluation. One method would be to compute ZCONT by solving for the unassigned y-variables as a linear programming problem. For many problems this would necessitate solving a large number of linear programming problems. However, by viewing each assignment to y as a change to the right hand side of the continuous LP, ZINT + ZCONT can be conveniently computed using sensitivity analysis.

7. COMPUTATIONAL EXPERIENCE

The performance of an integer programming algorithm is measured by its ability to solve a wide class of integer programming problems within reasonable computer time and storage limitations. To provide a basis for comparison with existing algorithms, the GIPC2 procedure was programmed in ANSI FORTRAN and implemented on the Purdue CDC 6500 computing system. Instructions for its use and a FORTRAN listing of the program are provided in reference [9].

The GIPC2 algorithm was evaluated on a set of seventeen linear and seventeen nonlinear test problems. Although the algorithm was specifically designed to solve the nonlinear integer programming problem, we were interested in evaluating the performance of the algorithm on linear problems as a special case. To provide a basis for comparing with existing linear integer programming algorithms, the seventeen linear test problems were also solved using a Branch and Bound code.

The main difficulty in comparing the computational efficiency of different integer programming algorithms is in developing a representative test problem set containing problems of varying size and difficulty. It is important to note that the relative performance of two integer programming algorithms may be highly dependent upon the test problem set used. In addition, it should be noted that problem size is only one factor in determining problem difficulty, and this factor is often dominated by problem structure. A problem with only five variables can be significantly more difficult to solve than a problem with twenty-five or more variables.

The set of linear test problems used in this investigation include four problems containing five variables each developed by Haldi [4], and thirteen additional problems of larger size. The four problems of Haldi, despite their small size, are difficult problems to solve and have been used extensively as a test bed for integer programming algorithms. Problem number five is a system design problem given by Petersen [10] and contains fourteen integer variables. The remaining twelve problems were randomly generated and range in size from ten variables to twenty-five variables and differ widely in their difficulty to solve. None of the test problems have explicit upper bounded variables.

The Branch and Bound code used in the investigation is the MINT mixed integer programming algorithm [6] based on the BBMIP code developed by the IBM Corporation [5] for the IBM 360 models 25 and above. The program is written in FORTRAN and is based upon the Dakin improved procedure of Land and Doig. A more modern code such as MPSX-MIP/360 or UMPIRE was not available on the Purdue CDC system, or it would have been used for a more meaningful comparison.

The computation times for the seventeen linear test problems are presented in Table 1. All times are in seconds and are for the Purdue CDC 6500 computing system. Times given in the table that are preceded by a greater-than sign indicate that the respective algorithm terminated without an optimal solution established after that amount of computation time.

TABLE 1. *Computational Experience — Linear*

Problem Number	Number of Constraints	Number of Variables	Computation Time (secs)	
			GIPC2	MINT
1	4	5	.545	4.099
2	4	5	.400	2.972
3	6	5	.608	3.375
4	6	5	.434	3.457
5	8	14	7.453	36.297
6	5	10	.811	20.221
7	5	10	1.042	21.001
8	10	10	.804	.537
9	10	10	.888	1.488
10	5	20	4.195	>188.
11	5	20	3.803	30.250
12	10	20	10.261	32.882
13	10	20	10.430	3.549
14	5	25	7.422	>188.
15	5	25	5.610	30.758
16	10	25	21.545	32.364
17	10	25	64.497*	5.440

*Reduced to 13.3 seconds by reordering variables in ascending order of their domain (see suggested modification in Section 8 below).

The GIPC2 code clearly outperformed the MINT code in solving the test problems of Haldi. Note that the MINT code required more time to solve problem 1 of Haldi containing only five variables than it required to solve problem 13 containing twenty variables. The test problems of Haldi clearly illustrate that problem structure can be more significant in determining problem difficulty than problem size.

In test problems 5 through 17, neither algorithm computationally dominates the other. The results for test problems 5, 6, 7, 9, 11, 12, 15, and 16 tend to indicate that the GIPC2 code is an average of 8.7 times faster than the MINT code, and the performance in problems 10 and 14 shows GIPC2 vastly superior. However this conclusion is contradicted by the results of test problems 8, 13, and 17 where the MINT code is 2.3 times faster than the GIPC2 code. The performance of the GIPC2 and MINT codes on these problems illustrates the unpredictable performance that is associated with integer programming algorithms.

A significant point of superiority of the GIPC2 code, however, is illustrated by comparative results on problems 10 and 14. Although the Branch and Bound procedure has been employed successfully to solve a number of large problems [2], it is sometimes misled into taking the wrong path early in the search. As a consequence, the Branch and Bound procedure can require an excessive amount of computer time to solve even relatively small problems. The MINT code failed to solve problems 10 and 14 after 188 seconds of computation. The computational results to date tend to indicate that the GIPC2 algorithm is less susceptible to getting sidetracked with large running times.

The performance of GIPC2 algorithm in solving integer programming problems with separable nonlinear objective functions was investigated by solving a set of seventeen nonlinear test problems. The nonlinear test problems were generated by using the constant arrays from the linear problem set with five or more of the linear terms in the objective function being replaced by nonlinear terms. Problems 18 through 21 each contain five variables and were

constructed from the problems of Haldi by replacing the linear objective function with five nonlinear stages. Problem 22 is a nonlinear version of the system design problem given by Petersen. The remaining twelve problems each contain from twenty-five to fifty variables and either five, fifteen, twenty, twenty-five, forty, or fifty nonlinear stages.

Computation times for the seventeen nonlinear test problems are given in Table 2. The computation times compare favorably with computation times for linear problems of similar size. Note that test problems 32 and 34, containing forty and fifty nonlinear stages respectively and ten constraints, each solved in less than twenty seconds. The data tends to suggest that the integer programming problem with nonlinear objective function is of relatively the same difficulty for the GIPC2 algorithm as the linear integer programming problem. The ability of the GIPC2 algorithm to solve the integer programming problem containing a separable nonlinear objective function in roughly equivalent times to that required to solve the linear integer programming problem is one of the primary contributions of the research reported here.

TABLE 2. *Computational Experience — Nonlinear*

Problem Number	Number of Constraints	Number of Variables	Number of Nonlinear Stages	GIPC2 Computation Time (secs)
18	4	5	5	.202
19	4	5	5	.212
20	6	5	5	.203
21	6	5	5	.215
22	8	14	12	10.312
23	10	25	5	4.905
24	10	25	5	15.494
25	10	25	15	7.550
26	10	25	15	20.204
27	10	25	20	6.998
28	10	25	25	5.522
29	10	25	25	7.299
30	10	25	25	9.756
31	10	40	25	12.493
32	10	40	40	11.892
33	10	50	25	17.596
34	10	50	50	18.717

The application of the nonlinear capability of the GIPC2 algorithm to a practical problem is illustrated by test problem 22, the nonlinear version of problem 5. In the original problem the system maintenance and operating costs which are to be minimized were assumed to be a linear function of the number of system components by type. However, in many systems the maintenance and operating costs are a nonlinear function of the number of system components by type. The restrictive linearity assumption is imposed primarily as a consequence of the lack of practical algorithms for solving the nonlinear integer programming problem. However the GIPC2 algorithm solved the more descriptive nonlinear version of the systems design problem in 10.312 seconds as compared to 36.297 seconds required by the MINT code to solve the linear version of the problem.

A major difficulty encountered in evaluating GIPC2 for solving nonlinear integer programming problems is in verifying that the solutions obtained are indeed optimal. The nonlinear

test problems are difficult problems to solve and alternate methods of solution apparently do not exist. To verify the accuracy of the GIPC2 algorithm in solving the nonlinear integer programming problem, a relatively simple ten-variable, five-constraint, nonlinear integer programming problem was exhaustively enumerated. The enumeration required approximately thirty-five minutes of computation time on the Purdue CDC 6500. The problem was solved by the GIPC2 algorithm yielding the same solution in approximately two seconds.

8. CONCLUSIONS

The generalized implicit enumeration scheme described in this paper can solve both linear and nonlinear integer programming problems. Computational experience indicates that the presence of nonlinearities has little or no affect on the computational efficiency of the algorithm. This attribute of the GIPC2 algorithm should allow for the formulation and solution of integer programming problems which fully consider the economies to scale which exist in the world.

A modification which would facilitate the use of the GIPC2 algorithm in solving larger problems is the replacement of the present simplex subroutine with a revised simplex method possessing implicit upper bounding procedures for the variables. This would allow the initial data matrix of the problem to be stored in external storage and would also avoid the need for inclusion of explicit upper bound constraints on the variables. This last feature would be particularly useful in solving zero-one integer programming problems.

A simple modification to the GIPC2 code that would result in considerably reduced computation time consists of incorporating a scheme for automatically reordering the variables in ascending magnitude of their domain prior to generating the set of candidate solutions. As a consequence of this reordering, the trees of candidate solutions would tend to be sparse near the top (Stage 1). As a result the number of partial evaluations examined would be reduced. The effect of this modification is illustrated by test problem 17 which originally required 64.5 seconds to solve. After manually reordering the variables in ascending order of their domain, the problem was solved in 13.3 seconds.

REFERENCES

[1] Balas, E., "An Additive Algorithm for Solving Linear Programs with Zero-One Variables," Operations Research, Vol. 13, pp. 517-546 (1965).

[2] Forrest, J.J.H., Hirst, J.P.H. and Tomlin, J.A., "Practical Solution of Large Mixed Integer Programming Problems with UMPIRE," Management Science, 20, No. 5, pp. 736-773 (1974).

[3] Geoffrion, A., "Objective Function Approximations in Mathematical Programming," Discussion Paper No. 61, Management Science Study Center, University of California, LA (May 1976).

[4] Haldi, J., "25 Integer Programming Test Problems," Working Paper No. 43, Graduate School of Business," Stanford University (December 1964).

[5] IBM Catalog of Programs for IBM System 360 Models 25 and Above, GC 20-1619-8, Program Number 360D-15.2.005.

[6] Kuester, J. and Mize, J., *Optimization Techniques with FORTRAN* (McGraw-Hill, 1973).

[7] Marsten, R. and Morin, T., "A Hybrid Approach to Discrete Mathematical Programming," Sloan School of Management, Working Paper 838-76 (March 1976).

[8] Morin, T. and Marsten, R., "An Algorithm for Nonlinear Knapsack Problems," Management Science, Vol. 22, No. 10 (1976).

[9] Pegden, C.D., "An Implicit Enumeration Algorithm for Solving Integer Programming Problems with Linear or Nonlinear Objective Functions," Ph.D. Dissertation, Purdue University (August 1975).

[10] Petersen, C.C., *Systems Planning and Evaluation Techniques*, Textbook in preparation.

[11] Plane, D.R. and C. McMillan, *Discrete Optimization* (Prentice-Hall, Inc., New Jersey, 1971).

[12] Taha, H., "A Balasian-Based Algorithm for Zero-One Polynomial Programming," Management Science, Vol. 18, No. 6 (1972).

[13] Watters, L.J., "Reduction of Integer Polynomial Programming Problems to Zero-One Linear Programming Problems," Operations Research 15, 1171-1174 (1967).

DUALITY FOR QUASI-CONCAVE PROGRAMS WITH APPLICATION TO ECONOMICS

T. R. Jefferson, G. M. Folie, and C. H. Scott

University of New South Wales
Kensington, N.S.W., Australia

ABSTRACT

A duality theory is developed for mathematical programs with strictly quasi-concave objective functions to be maximized over a convex set. This work broadens the duality theory of Rockafellar and Peterson from concave (convex) functions to quasi-concave (quasi-convex) functions. The theory is closely related to the utility theory in economics. An example from economic planning is examined and the solution to the dual program is shown to have the properties normally associated with market prices.

1. INTRODUCTION

Although duality theory for linear programming has been well developed and widely used for some time, it is only in recent years that significant advances have been made in duality theory for convex (concave) programs. Notable contributions have been made by Rockafellar [13] and Peterson [12]. Despite these developments, there are still many programming problems that are not encompassed by the existing theoretical developments. One such important class of mathematical programs are quasi-concave programs, and it is the purpose of this paper to extend the benefits of duality theory to this class of programs.

In 1967, Arrow and Enthoven [1] developed necessary and sufficient conditions for the optimality of quasi-concave programs. Later Luenberger [11] developed a duality theory for quasi-concave programs, which separated primal and dual variables. This duality theory was expanded by Greenberg and Pierskalla into surrogate duality [5], [7].

The duality theory developed here is valid for quasi-concave programs with closed strictly quasi-concave objective functions. This work is motivated by the dual relationship between goods and prices first observed by Roy [14] and by Peterson's work in duality theory [12]. The major result of this paper, lies in the separation of the objective function from a linear constraint set, which simplifies the derivation of the dual program, as well as the relationship between the primal and dual programs. Furthermore, the duality theory developed here is widely applicable to a class of problems found in economics.

Duality theory comes naturally to linear programs via Kuhn-Tucker theory and the linearity of the problem. In general, the existence of non-linearities in mathematical programs raises a number of problems and makes generalizations more complex and difficult. Wolfe duality is an example of this problem. For concave programs, the concave conjugate transform can be

used to derive a dual program, and to develop all the associated primal-dual relationships (Peterson [12]).

In order to clarify the difference between the duality theory developed in this paper, and that currently used for concave programs, as well as why a special duality theory is needed, the following brief digression will be made. Consider the following definitions.

DEFINITION: The concave conjugate transform of a function $g(x)$ defined on a set C is the pair $[h, D]$ defined by

$$h(y) \triangleq \inf_{x \in C} \{< x, y > - g(x)\}$$

$$D \triangleq \{y \mid \inf_{x \in C} < x, y > - g(x) > - \infty\}.$$

DEFINITION: The hypograph of a function $g(x)$ is the set:

$$\{(x, \beta) \mid x \in C, \ \beta \leqslant g(x)\}.$$

DEFINITION: A concave (quasi-concave) function is closed when its hypograph is a closed set.

DEFINITION: The supergradient of a function g at a point x is the set $\partial g(x)$ defined by

$$\partial g(x) = \{y \mid g(x) + < y, z - x > \geqslant g(z), \ \forall z \in C\}.$$

By construction $h(y)$ is a closed concave function. In addition for $x \in C$ and $y \in D$ we have the following inequality:

(1) $$g(x) + h(y) \leqslant < x, y >.$$

The inequality (1) is an equality when

$$x \in \partial h(y) \quad \text{or} \quad y \in \partial g(x).$$

The concave conjugate transform generates very strong relationships. If $g(x)$ is not concave, the concave conjugate transform still operates on $g(x)$ as if it were concave. The conjugate transform of a non-concave function, $g(x)$, does not use the hypograph of $g(x)$, but only the convex hull of the hypograph of $g(x)$. Thus some information regarding the properties of $g(x)$ is lost by this transform.

This undesirable feature of the conjugate transform, indicates the need to develop a new transform, which can be used to derive a duality theory for non-concave programs.

At first, it may not appear to be a particularly serious limitation that the conjugate transform cannot be used with non-concave functions. However there are cases where a transform that will handle quasi-concave programs is required. For instance, the conjugate transform cannot be used to derive dual programs for problems in economic theory. The reason is that economic theorists have, over the years, reduced the restrictiveness of both consumer and producer theory. The fundamental property of the utility function in the theory of consumer behaviour, which stems from the axioms of weak preference ordering, is that indifference curves, or constant utility surfaces define convex sets. Equivalently, economists

require these indifference curves to have the property of diminishing marginal rates of substitution (Green [4]). Thus the minimal property of any utility function, used to represent consumer choice behaviour, is quasi-concavity. Thus in order to derive any dual programs for economic problems, it is essential that the transform used to obtain the dual is valid for quasi-concave programs.

The transform derived here has these desired properties, and will be called the utility transform.

The properties of the utility transform are derived in the next section and are then used to develop a duality theory for quasi-concave programming. This theory is an extension of the duality theory developed by Luenberger [11]. An example from economics is presented at the end of the paper in order to elucidate the usefulness of the utility transform.

2. THE UTILITY TRANSFORM

A form of duality between prices and commodities in consumer theory was originally noted by Roy [14] in 1947. Roy's work explicitly developed a dual relationship between the consumer's direct utility function, which is a function of his commodity bundle, and an indirect utility function, which is a function of the prices of the commodities and consumer income. Recently this theory has been used to provide a clearer understanding of consumer theory by proving, in a simple manner, a large number of propositions in consumer theory. Lau [10] and Diewert [3] provide a useful compendium of this work. Although the concept of the utility transform stems from the relationship between the direct and indirect utility functions, the purposes of this paper require the development of a slightly different approach to duality. Consider the pair $[u, U]$, of utility function $u(x)$ defined on the convex set U.

DEFINITION: The utility transform of $[u, U]$ is a pair $[\mathrm{v}, V]$ defined by

$$\mathrm{v}(p) \triangleq \inf_{x \in U} \{-u(x) \mid < p, x > \leqslant 0\}$$

$$V \triangleq \{p \mid \inf_{x \in U} [-u(x) \mid < p, x > \leqslant 0] > -\infty\}.$$

By construction, for $x \in U$, $p \in V$ and $< p, x > \leqslant 0$ we have the utility inequality holding:

$$u(x) + \mathrm{v}(p) \leqslant 0.$$

The construction of $\mathrm{v}(p)$ and the indirect utility function differ in the following ways. Firstly the linking constraint

$$< p, x > \leqslant 0$$

is generally referred to as the budget constraint, but usually in consumer theory, it has a positive right hand side. However, as will be seen later, it is convenient to absorb the right-hand side into the inner product. This can be done by identifying consumer income as another commodity, which although the consumer has an endowment of it, does not wish to have it for its own sake. A more general interpretation, is to consider x to measure the quantity of goods and services that a consumer buys and sells. We use this second approach in the example (section 4). Finally we take the infimum of $-u(x)$ rather than the supremum of $u(x)$. These differences allow us to work with quasi-concave functions. $\mathrm{v}(p)$ is quasi-concave, whereas the indirect utility function is quasi-convex. More importantly though, the absorption of the right-hand side of the budget constraint permits a complete separation of the price and commodity variables.

It is accepted that the utility transform is not the only method for handling duality. Greenberg and Pierskalla [5] developed a surrogate dual, which in terms of the notation used here, is defined by

$$\inf_{x \in U} \{- u(x) \mid \sum_{i=1}^{m} p_i(g_i(x) - b_i) \leqslant 0\}$$

where $g_i(x) \leqslant b_i$ is a constraint to be satisfied. Clearly, when $g_i(x) - b_i$ is replaced by x_i, then the above expression is the same as the utility inequality.

The surrogate dual was further specialized for quasi-concave functions in [7] by Greenberg and Pierskalla to the z-quasi-conjugate

$$u_z^*(p) = z + \inf \{- u(x) \mid < x, p > \leqslant z\}.$$

This becomes the utility transform when $z = 0$. In their paper on "Quasi-Conjugate Functions and Surrogate Duality" [7], Greenberg and Pierskalla develop the properties of the z-quasi-conjugate. This paper formed the basis of a further analysis by Crouzeix into the properties of quasi-concave functions [2].

The z variable in the z-quasi-conjugate is difficult to handle in the dual. In order to take cone conditions into consideration, it is necessary to have $< x, p > \leqslant 0$. This means that $z = 0$, and we are left with the more convenient utility transform.

We now develop the properties of [v, V]. In order to do this we require a relaxation of the concept of supergradient.

DEFINITION: The local supergradient of a quasi-concave function $u(x)$, $x \in U$ is a set $\partial^{\text{loc}} u(x)$ defined by:

$$\partial^{\text{loc}} u(x) \triangleq \{p \mid \lim_{\alpha \to 0} \frac{u(x + \alpha\ \Delta x) - u(x)}{\alpha} \leqslant < \Delta x, p > \forall\ \Delta x$$

$$\text{such that } x + \beta\ \Delta x \in U, \beta > 0\}.$$

The local supergradient is a generalization of the concept of supergradient presented earlier. For the usual case of differentiable functions, the local supergradient contains a single element: the gradient. It is through the local supergradient that [u, U] and [v, V] can be related.

It is necessary to know this relationship in order to develop the duality theory presented in the next section. The relationship between [u, U] and [v, V] is formally expressed by Theorem 1, which is stated at the end of this section and proved in the Appendix. In order to prove Theorem 1, the following four Lemmas are needed.

LEMMA 1: $v(p)$ is quasi-concave and positively homogeneous of degree zero. V is a convex cone. $\partial^{\text{loc}} v(p)$ is positively homogeneous of degree minus one.

PROOF: See appendix.

LEMMA 2: For v, the utility transform of u, hypo v is closed if hypo u is closed.

PROOF: See appendix or [7].

LEMMA 3: For u closed, $p \in V$, $x \in U$ and $< p, x > \leqslant 0$ we have the utility inequality

(2) $$u(x) + v(p) \leqslant 0.$$

When equality holds in (2) we have:

(i) $\lambda p \in \partial^{loc} u(x)$, $\lambda \geqslant 0$

(ii) $\mu x \in \partial^{loc} v(p)$, $\mu \geqslant 0$.

PROOF: See appendix.

LEMMA 4: Suppose u is a closed strictly quasi-concave function on U with utility transform $[v, V]$.

Then either

(i) The maximum of u is attained at a point $z \in U$ and $v(p) = -u(z)$ for $p \in V \cap \{p \mid < p, z > \leqslant 0\}$. Let p be equivalent to q ($p \equiv q$) if there exists $\alpha > 0$ such that $p = \alpha q$. $v(p)$ is a strictly quasi-concave function on $V \cap \{p \mid < p, z > \geqslant 0\}$ with respect to the quotient space defined by this equivalence relation, or

(ii) The supremum of $u(x)$ is infinity. $v(p)$ is a strictly quasi-concave function on V with respect to the quotient space defined in (*i*). See appendix for proof.

THEOREM 1: Let $[u, U]$ have the following properties:

(i) the hypograph of u is closed.

(ii) $u(x)$ is strictly quasi-concave.

(iii) $[v, V]$ is the utility transform of $[u, U]$.

(iv) $z \in U$ is the optimal point of $\sup_{x \in U} \{u(x)\}$ if it exists.

Given that $x \in U$, $p \in V$ and $< p, x > \leqslant 0$, $x \not\equiv z$ then: $u(\bar{x}) + v(\bar{p}) = 0$ if and only if either

(I) $$\lambda \bar{p} \in \partial^{loc} u(\bar{x}), \quad \lambda > 0$$

or

(II)
(a) $\bar{p} \in S = V$, or $V \cap \{p \mid < p, z > \geqslant 0\}$ if z exists

(b) $\nu \bar{x} \in \partial^{loc} v(\bar{p})$, $\nu > 0$

(c) $< \bar{p}, \bar{x} > = 0$

(d) $u(\bar{x}) = \sup_{y \in U} \{u(y) \mid y \equiv \bar{x} \text{ or } y \equiv -\bar{x}\}$

See the appendix for proof of theorem.

3. DUALITY THEORY

Consider the following quasi-concave program.

PROGRAM A max. $u(x)$
subject to $x \in U \cap \chi$

where u is a closed strictly quasi-concave function defined on the convex set U, and χ is a convex cone. We assume that if z is such that $u(z) = \sup_{x \in U} u(x)$ then $z \notin \chi$, together with $U \cap \chi \neq \phi$ and $\sup \{u(x) \mid x \in U \cap \chi\} < \infty$. The economic dual to Program A is:

PROGRAM B max. $v(p)$
subject to $p \in V \cap \Pi$

where $[v, V]$ is the utility transform of $[u, U]$, and Π is the dual cone to χ defined by,

$$\Pi = \{p \mid < p, x > \leqslant 0, \forall x \in \chi\}.$$

At optimality the following relations hold:

$$\bar{x} \in U \cap \chi, \ \bar{p} \in V \cap \Pi$$
$$< \bar{p}, \bar{x} > = 0$$
$$u(\bar{x}) + v(\bar{p}) = 0$$
$$\lambda \bar{p} \in \partial^{loc} u(\bar{x}), \ \lambda > 0$$
$$\nu \bar{x} \in \partial^{loc} v(\bar{p}), \ \nu > 0$$
$$u(\bar{x}) = \sup_{y \in U} \{u(y) \mid y \equiv \bar{x} \text{ or } y \equiv -\bar{x}\}.$$

An interpretation of these optimality conditions will be given, when the example is discussed in the following section.

THEOREM 2: The previously mentioned optimality conditions are necessary and sufficient for optimality for Programs A and B.

4. *EXAMPLE*

The theory of economic planning is concerned with devising an allocation of resources which maximizes a given social welfare function of the society in question, given that the society has a prescribed quantity of endowments of labour, together with some consumption goods remaining from the previous period. The society has available a set of known production technologies, which take inputs, such as labour, and transform them into consumer goods. The use of some goods as both production inputs and consumer goods is not precluded. Problems of this type are discussed by Heal [8].

Thus in a directive economy, the central planning office must effectively solve a large mathematical programming problem. In order to illustrate the duality concepts developed earlier, consider a simplified version of the planning problem which contains all the essential elements of economic planning. The problem is defined as follows:

$$\max\ w(x + e) \triangleq w(x;\ e)$$
$$\text{subject to } x \geqslant -e$$
$$x = Az;\ z \geqslant 0$$
$$\text{where} \quad x = \begin{bmatrix} x^c \\ x^l \end{bmatrix}.$$

lere, superscripts denote vectors and subscripts denote scalars.

$w(x;\ e)$ is a known quasi-concave social welfare function which captures the preference rderings that this society has for its consumption of goods, $x^c + e^c$, and the use of leisure, $^l + e^l$. The society has an endowment of leisure, e^l which it can forego, in quantities x^l to rovide labour for productive activities to produce more consumer goods, x^c, which increase ggregate social welfare. It should be noted that the nature of the coefficients of the production ctivity matrix, A, will ensure that only positive quantities of consumer goods will be produced, nd that leisure will be consumed by production, $x^l \leqslant 0$; i.e. only positive quantities of labour ill be supplied.

This problem will now be expressed in terms of the format developed in Section 3 in rder to illustrate the relationship between the primal and dual programs. This will be done for ocial welfare functions which are assumed to be log-linear. The problem can now be expressed a form similar to that referred to as PROGRAM A in the previous section:

$$\max\ w(x;\ e) = \sum_i \alpha_i \log(x_i + e_i)$$
$$\text{subject to } x \in U \cap \chi$$
$$\text{where } U = \{x \mid x_i \geqslant -e_i,\ \forall i\}$$
$$\text{and, } \chi = \{x \mid x = Az;\ z \geqslant 0\}.$$

learly χ is a cone, which is defined by the technological possibilities available to the economy. hus, this particular formulation of the primal problem treats production through the cone, hich, as will be shown below, results in a considerable simplification.

To obtain PROGRAM B, the economic dual, it is necessary to derive the utility transform $w(x;\ e)$:

$$\inf_{x \in U} \{-w(x;\ e) \mid < p,\ x > \leqslant 0\}.$$

orming the Lagrangian, and then differentiating with respect to x_i and λ, the following results e obtained:

$$\alpha_i(x_i + e_i)^{-1} - \lambda p_i = 0 \quad \forall i$$
$$< p,\ x > = 0.$$

From simple algebra, it can now be shown that

$$\lambda = \frac{\sum \alpha_i}{\sum p_i e_i}$$

and

$$x_i = -e_i + \frac{\alpha_i}{\sum \alpha_i} \frac{\sum p_j e_j}{p_i} \quad \forall i.$$

Before continuing, the discussion can be simplified, by assuming that the α_i are selected so the $\sum \alpha_i = 1$, since all the expressions derived clearly imply that the α_i are normalized.

Substituting for x_i, the utility transform $v(p; e)$ of $w(x; e)$ is obtained:

$$v(p; e) = \sum_i \alpha_i \log \left[\frac{1}{\alpha_i} \frac{p_i}{\sum_j p_j e_j} \right]$$

$$V = \{p \mid p_i \geqslant 0, \ \forall i\}.$$

Thus, the economic dual to the original problem can now be written in the form of PROGRAM B given in the previous section:

$$\max \sum_i \alpha_i \log \left[\frac{1}{\alpha_i} \frac{p_i}{\sum p_j e_j} \right]$$

$$\text{subject to } p \in V \cap \Pi$$

$$\text{where } V = \{p \mid p_i \leqslant 0, \quad \forall i\}$$

$$\text{and } \Pi = \{p \mid A' p \leqslant 0\}$$

where Π is the polar cone of χ and its derivation follows from the well known properties of finite cones.

It is of some interest to provide an interpretation of this dual program. The most important property that emerges is that the dual program is expressed solely in terms of the dual variables, p, while retaining all the basic parameters that defined the primal program. It needs little appeal to one's intuition to interpret the dual variables, p, as some type of price vector. Unfortunately this, in itself, does not provide much insight, since the dual programs to various problems, for example, linear programming and posynomial programming, generate dual variables with quite different interpretations.

For dual programs derived by use of the utility transform, the key lies in the requiremen that at optimality $< p, x > = 0$. The optimizing process implicit in the utility transform is identical to the well known optimizing problem in economics in which a consumer selects his most preferred commodity bundle, subject to the restriction that his expenditure must no exceed his income. Thus, it seems plausible to interpret the dual variables, p, as market prices A careful examination of, not only the dual program, but also the relationships between the primal and dual variables at optimality, indicates that this is a valid and useful interpretation.

As a consequence of the utility transform [v, V], the dual variables, p, must be non negative, which is a requirement for a sensible price system.

The dual variables, p, are related to the primal variables, x, by the relationship

$$\lambda \bar{p}_i = \frac{\partial w(\bar{x}; e)}{\partial x_i}$$

The Lagrange multiplier, λ, from the utility transform appears, because even if the optimal solution to the primal problem, $\bar{x}$, is known the magnitudes of the resulting prices depend on the units of measurement used in $w(x; e)$.

This relationship indicates that in order to have an absolute measure of the dual variables, $\bar{p}$, when the solution, $\bar{x}$, to the primal problem is known, it is necessary to know, λ, which arises in the utility transform. By examination of the utility transform, it is clear that the Lagrange multiplier, λ, can be interpreted as the marginal utility of society's endowment, and its magnitude clearly depends on the units in which the social welfare, w, is expressed. In the planning problem considered here, it can easily be seen that $\lambda = 1/\sum_j p_j e_j$, which is the reciprocal of the value of society's endowment. Furthermore, as λ is neither a primal nor a dual variable, it is a linking variable. As indicated earlier, consumer preferences for different commodity bundles can be expressed as an ordinal function, which need only be quasi-concave. If it were accepted that utility could be measured absolutely, and was not merely an ordering concept, then one could assume a concave utility function and then use a duality theory based on the conjugate transform. Similar comments are valid for the social welfare function, w, which is also ordinal.

If commodity $i = 1$, is designated as the numeraire good, then a set of relative prices can be used to define the optimality condition

$$\frac{p_i}{p_1} = \frac{\partial w(x; e)}{\partial x_i} \Big/ \frac{\partial w(x; e)}{\partial x_1} \quad \forall i.$$

Thus, at optimality, the familiar result emerges; namely that the relative price of good i (in terms of good 1) is equal to the marginal rate of substitution of good i for good 1, MRS_{i1}. A close examination of the dual social welfare function, $v(p; e)$ indicates that it is similar to the indirect utility referred to earlier. The difference here is that the dual social welfare function has the term, $\sum_j p_j e_j$, which is clearly the market value of society's endowment given a price vector p. This is similar to the notion of income, which is used in conventional consumer theory. As an aside, it should be noted that the indirect utility function as used by Lau [10] in consumer theory can be expressed in the form of $v(p; e)$ if consumer income, Y, is assumed to be the only endowment, with a price of 1, and all other goods are purchased by the sale of the endowment (income).

Keeping this discussion in mind, and accepting the interpretation of the dual variables, at optimality, as market prices, it is now possible to provide an insight into another optimality condition: $\nu \bar{x} \in \nabla v(p; e)$. It can be readily shown, $\nu = \sum_j p_j e_j$. This optimality condition results in a set of relationships similar to the demand equations found in consumer theory. For the planning problem being analysed here, this condition tells us that if an optimal solution to the dual is known, $\bar{p}$, then the quantity of consumer goods that will be produced, x^c, and the amount of leisure foregone, x^l, to produce these goods is obtained by this differential.

The condition of primal and dual feasibility, together with the requirement that the linking condition, $< \bar{p}, \bar{x} > = 0$, holds at optimality, can be used to develop some insights into the production sector.

Primal feasibility: $x \in U \cap \chi \Rightarrow x \geqslant -e$ and $x = Az$; $z \geqslant 0$.

Dual feasibility: $\bar{p} \in V \cap \Pi \Rightarrow \bar{p} \geqslant 0$ and $A'\bar{p} \leqslant 0$.

The linking condition, $< \bar{p}, \bar{x} > = 0$, is interpreted as a trade balance constraint which ensures that the value of the goods produced is equal to the value of the compensation paid for the inputs used to manufacture these goods.

By direct substitution, $< \bar{p}, \bar{x} > = < \bar{p}, A\bar{z} > = < A'\bar{p}, \bar{z} > = 0$.

From the primal and dual feasibility conditions, $\bar{z} \geqslant 0$ and $\bar{p} \geqslant 0$, this means that if $< A'\bar{p}, \bar{z} > = 0$, then each term of this scalar expression must be zero. This is only possible when for each production activity j:

$$\text{if } \sum_j a_{ij}\bar{p}_i < 0, \text{ then } \bar{z}_j = 0$$

$$\text{or if } \sum_j a_{ij}\bar{p}_i = 0, \text{ then } \bar{z}_j > 0.$$

Thus by use of the optimality conditions, derived in section 3, the familiar complementary slackness conditions emerge. These can be given the usual economic interpretation. If a particular activity j is included in the optimal plan, then $\bar{z}_j > 0$ and thus $\sum_i a_{ij}p_i = 0$, which means that there are no excess profits and the value of the labour (foregone leisure) used in this particular production activity is equal to the value of the output. On the other hand if a particular activity is uneconomic, $\bar{z}_j = 0$, then the value of the labour services used to operate the activity at unit level is greater than the returns from the products produced by the activity. Again, the results that emerge are the standard ones encountered in the economic theory of market behaviour.

Thus it can be seen that the planning problem formulated at the beginning of this section can be viewed in a completely different manner. An alternative solution to the original planning problem, which required the planners to issue the directives, $\bar{x}$, to everyone in society, would be to solve the dual program to obtain a set of prices, $\bar{p}$, which can be interpreted as market prices. Then the planners need only announce these prices, $\bar{p}$, and the members of society, by responding to these price signals will generate an allocation of resources equivalent to that which would have occurred under a directive, $\bar{x}$.

ACKNOWLEDGMENT

The authors wish to thank the referee for his helpful comments and for pointing out the important paper [7].

BIBLIOGRAPHY

[1] Arrow, K.J. and A.C. Enthoven, "Quasi-Concave Programming," Econometrica, *29*, 779-800 (1967).

[2] Crouzeix, J.P., "Contributions à l'étude des fonctions quasi-convexes," (Ph.D Dissertation, University of Clermont, France, 1977).

[3] Diewert, W.E., "Applications of Duality Theory" in *Frontiers of Quantitative Economics*, Vol. II, M.D. Intriligator and D.A. Kendrick, Editors (American Elsevier Publishing Co., New York, 1974).

[4] Green, H.A.J., *Consumer Theory*, (MacMillan, London, 1976).

[5] Greenberg, H.J. and W.P. Pierskalla, "Surrogate Mathematical Programming," Operations Research, *18*, 924-939 (1970).

[6] Greenberg, H.J. and W.P. Pierskalla, "A Review of Quasi-Concave Functions," Operations Research, *19*, 1533-70 (1971).
[7] Greenberg, H.J. and W.P. Pierskalla, "Quasi-Conjugate Functions and Surrogate Duality," Cahiers du Centre d'Etudes de Recherche Opérationelle, *15*, 437-448 (1973).
[8] Heal, G.M., *The Theory of Economic Planning*, (American Elsevier, New York, 1973).
[9] Jefferson, T.R., G.M. Folie and C.H. Scott, "Dual Games," School of Mechanical and Industrial Engineering Report (1977).
[10] Lau, L.J., "Duality and the Structure of Utility Functions," Journal of Economic Theory, *1*, 374-396 (1969).
[11] Luenberger, D.G., "Quasi-Convex programming," SIAM Journal on Applied Mathematics, *16*, 1090-1095 (1968).
[12] Peterson, E.L., "Geometric Programming," SIAM Review, *18*, 1-52 (1976).
[13] Rockafellar, R.T., *Convex Analysis* (Princeton University Press, Princeton, New Jersey, 1970).
[14] Roy, R., "La distribution du revenu entre les divers biens," Econometrica, *15*, 205-225 (1947).

APPENDIX

PROOF OF LEMMA 1: Consider any two points p_1 and p_2 and $0 < \lambda < 1$.

$$v(\lambda p_1 + (1-\lambda)p_2) = \inf_{x \in U} \{-u(x) \mid < \lambda p_1 + (1-\lambda)\, p_2,\ x > \leqslant 0\}$$

$$\geqslant \min\left[\inf_{x_1 \in U} \{-u(x_1) \mid < p_1,\ x_1 > \leqslant 0\},\ \inf_{x_2 \in U} \{-u(x_2) \mid < p_2,\ x_2,\ > \leqslant 0\}\right]$$

$$= \min\left[v(p_1),\ v(p_2)\right]$$

This proves the quasi-concavity of $v(p)$.

Consider $\lambda > 0$, then

$$v(\lambda p) = \inf_{x \in U} \{-u(x) \mid < \lambda p,\ x > \leqslant 0\}$$

$$= \inf_{x \in U} \{-u(x) \mid < p,\ x > \leqslant 0\} = v(p).$$

Thus $v(p)$ is positively homogeneous of degree zero.

V is a convex cone by construction.

Let $x \in \partial^{\text{loc}} v(\gamma p)$, $\gamma > 0$. By definition

3) $$\lim_{\alpha \to 0} \frac{y(\gamma p + \alpha\, \Delta p)}{\alpha} \leqslant < \Delta p,\ x >.$$

Since $v(\gamma p)$ is positively homogeneous of degree zero, (3) implies

4) $$\lim_{\alpha \to 0} \frac{v(p + \alpha \frac{\Delta p}{\gamma}) - v(p)}{\alpha} \leqslant < \Delta p,\ x >.$$

Substituting in (3) $\Delta q = \dfrac{\Delta p}{\gamma}$ we obtain

$$\frac{v(p + \alpha\, \Delta q) - v(p)}{\alpha} \leqslant < \Delta q,\ \gamma x > . \tag{5}$$

Thus by (5) we have proven that $x \in \partial^{\text{loc}} v(\gamma p)$ if and only if $\gamma x \in \partial^{\text{loc}} v(p)$. Similar properties are proved for the surrogate dual in [5].

PROOF OF LEMMA 2: By definition, we have that

$$\text{hypo } v = \{(p,\ \beta) \mid p \in V,\ \beta \leqslant v(p)\}.$$

Let $\{(p_i,\ \beta_i)\}$ be a convergent series with

$$\lim_{i \to \infty} (p_i,\ \beta_i) = (\bar{p},\ \bar{\beta}) \text{ and } (p_i,\ \beta_i) \in \text{hypo } v, \text{ for all } i.$$

We require to show that $(\bar{p},\ \bar{\beta})$ is a member of hypo v. Assume that $(\bar{p},\ \bar{\beta}) \notin$ hypo v. There are two possibilities to consider: (i) $\bar{p} \notin V$ and (ii) $\bar{\beta} > v(\bar{p})$. For case (i), $\lim_{i \to \infty} \beta_i = -\infty$ by definition. This contradicts the assumption that $\{(p_i,\ \beta_i)\}$ is convergent. Hence $\bar{p} \in V$. In case (ii), we let $(x_i,\ \alpha_i)$ be such that

$$v(p_i) = -u(x_i) = -\alpha_i,\ \beta_i \leqslant \alpha_i\ \forall i.$$

This is admissable by the definition of V. We let

$$\lim_{i \to \infty} (x_i,\ \alpha_i) = (\bar{x},\ \bar{\alpha})$$

where $\{(x_i,\ \alpha_i)\}$ and $(\bar{x},\ \bar{\alpha})$ belong to hypo u; the latter since hypo u is closed. Hence $\bar{\alpha} = u(\bar{x})$ and $v(\bar{p}) = -u(\bar{x})$. This in turn implies that $\bar{\beta} \leqslant v(\bar{p})$, which is a contradiction.

Hence hypo v is closed. A similar lemma is proved in [7].

PROOF OF LEMMA 3: By the utility transform we have

$$v(p) = \inf_{x \in U} \{-u(x) \mid < p,\ x > \leqslant 0\}. \tag{6}$$

The solution to (6) is the solution to the saddlepoint problem

$$v(p) = \inf_{x \in U} \sup_{\lambda \geqslant 0} \{\lambda < p,\ x > - u(x)\}.$$

The first order conditions require that

$$\lambda p \in \partial^{\text{loc}} u(x),\ \lambda \geqslant 0.$$

Consider now the transform of $v(p)$

$$\inf_{p \in V} \sup_{\mu \geqslant 0} \{\mu < p,\ x > - v(p)\}.$$

The first order conditions require:

$$\mu\, x \in \partial^{\text{loc}} v(p),\ \mu \geqslant 0.$$

PROOF OF LEMMA 4: Suppose that there exists $z \in U$ such that

$$u(z) = \sup_{x \in U} \{u(x)\} \text{ is defined}$$

$$v(p) = \inf_{x \in U} \{-u(x) \mid < p, x > \leqslant 0\}.$$

Thus $v(p) = -u(z)$ for $< p, z > \leqslant 0$

Consider $p_1, p_2 \in V \cap \{p \mid < p, z > \geqslant 0\}$ such that $p_1 \not\equiv p_2$ and $p_1 \not\equiv -p_2$.

Choose $0 < \lambda < 1$.

$$v(\lambda p_1 + (1-\lambda)p_2) = \inf_{x \in U} \{-u(x) \mid < \lambda p_1 + (1-\lambda)p_2, x > \leqslant 0\}$$

$$> \min \left[\inf_{x_1 \in U} \{-u(x_1) \mid < p_1, x_1 > \leqslant 0\}, \inf_{x_2 \in U} \{-u(x_2) \mid < p_2, x_2 > \leqslant 0\} \right]$$

$= \min [v(p_1), v(p_2)]$ by the strict quasi-concavity of u.

Suppose $\sup_{x \in U} \{u(x)\}$ is undefined.

Consider $p_1, p_2 \in V$ such that $p_1 \not\equiv p_2$.

Choose $0 < \lambda < 1$

$$v(\lambda p_1 + (1-\lambda)p_2) = \inf_{x \in U} \{-u(x) \mid < \lambda p_1 + (1-\lambda)p_2, x > \leqslant 0\}$$

$> \min [v(p_1), v(p_2)]$ by the strict quasi-concavity of u.

PROOF OF THEOREM 1: First assume

7) $$u(\bar{x}) + v(\bar{p}) = 0$$

(7) implies $\lambda \bar{p} \in \partial^{\text{loc}} u(\bar{x})$ and $\nu \bar{x} \in \partial^{\text{loc}} v(\bar{p})$ by Lemma 3. λ and ν are positive because $(\bar{x})$, $v(\bar{p})$ are strictly quasi-concave.

Suppose $\bar{p}$ satisfies $< \bar{p}, z > < 0$. Since $\bar{x} \not\equiv z$

8) $$\{p \mid < p, x > \leqslant 0, < p, z > > 0\} \neq \phi$$

(p) is strictly quasi-concave on this set. For p belonging to the set defined by (8), $(p) > v(\bar{p})$. This contradicts the utility inequality. Therefore (a) must hold.

Suppose $< \bar{p}, \bar{x} > < 0$ then by the strict quasi-concavity of $u(x)$ there exists on x such hat $< \bar{p}, x > = 0$ and $u(x) > u(\bar{x})$. This contradicts the utility inequality. Therefore (c) nust hold.

Suppose $u(\bar{x}) < \sup_{y \in U} \{u(y) \mid y \equiv \bar{x} \text{ or } y \equiv -\bar{x}\}$.

his too would contradict the utility inequality. Thus (d) must hold.

Now going the other way suppose we have $\lambda \bar{p} \in \partial^{\text{loc}} u(\bar{x})$.

Consider

$$v(\bar{p}) = \inf_{x \in U} \{-u(x) \mid < \bar{p}, x > \leqslant 0\}. \tag{9}$$

By property (i) if the infimum to (9) exists it is attained on $U \cap \{x \mid < \bar{p}, x > \leqslant 0\}$. Because $u(x)$ is strictly quasi-concave a local minimum is a global minimum. Therefore

$$v(\bar{p}) = -u(\bar{x}).$$

Suppose now (IIa-d) hold. $v(p)$ is strictly quasi-concave in the sense of Lemma 4 on S. Therefore (b) and Lemma 2 imply

$$-v(\bar{p}) = \inf_{p \in S} \{-v(p) \mid < p, \bar{x} > \leqslant 0\}.$$

Let $\bar{x}$ satisfy (d) in addition. Let $\lambda \hat{p} \in \partial^{\text{loc}} u(\bar{x})$, $\lambda > 0$. $\partial^{\text{loc}} u(\bar{x})$ is non-empty because of property (i).

Consider

$$\inf_{x \in U} \{-u(x) \mid < \hat{p}, x > \leqslant 0\}. \tag{10}$$

The infimum of (10) is attained at $\bar{x}$ by construction.

Therefore $v(\hat{p}) = -u(\bar{x})$ and $< \hat{p}, \bar{x} > \leqslant 0$. By construction $-v(\bar{p}) < -v(\hat{p})$ if $\hat{p} \not\equiv \bar{p}$. This implies $u(\bar{x}) + v(\bar{p}) > 0$ which contradicts the utility inequality. Therefore $v(\bar{p}) = b(\hat{p})$ and

$$u(\bar{x}) + v(\bar{p}) = 0.$$

Note that if $\bar{x} = z$ and $\bar{p}$ satisfying $< \bar{p}, \bar{x} > \leqslant 0$ and $\bar{p} \in V$ will satisfy the utility inequality. λ and ν are equal to zero and we lose the relationships I and II between the primal and dual variables. The primal problem reduces to one of global minimization which is relatively straightforward.

PROOF OF THEOREM 2: Assume $\bar{x}$ is in optimum for Program A. Thus there exists a vector p such that $\lambda p \in \partial^{\text{loc}} u(x)$, $\lambda > 0$. $< \lambda p, \Delta x > \leqslant 0$ for $\Delta x \in \chi$. Since $\bar{x} \in \chi$, $< \lambda p, \bar{x} > \leqslant 0$. Thus $p \in \Pi \cap V$ by construction and Theorem 1. Also by Theorem 1

$$u(\bar{x}) + v(p) = 0.$$

Also by Theorem 1 the remaining conditions hold since $< p, \Delta x > \leqslant 0 \ \forall \Delta x \in \chi$. Now suppose the optimum for Program B is $p^* \in V \cap \Pi$ such that $v(p^*) > v(p)$. This would contradict $u(\bar{x}) + v(p^*) \leqslant 0$ which is impossible or imply $< \bar{x}, p^* > > 0$. This too is impossible since $\bar{x} \in \chi$ and $p^* \in \Pi$. Therefore p is optimal for Program B.

Suppose now we have $\bar{p}$ and optimum for Program B. Thus there exists a vector x such that $\nu x \in \partial^{\text{loc}} v(\bar{p})$ $\nu > 0$, $u(x) = \sup_{y \in U} \{u(y) \mid y \equiv x \text{ or } y \equiv -x\}$ and $< x, \Delta p > \leqslant 0$ for $\Delta p \in \Pi$. Since $\bar{p} \in \Pi$, $< x, \bar{p} > \leqslant 0$. By construction and Theorem 1 $x \in \chi \cap U$. Also by Theorem 1

$$u(x) + v(\bar{p}) = 0,$$

and the remaining optimality conditions hold.

Suppose the optimum for Program A is $x^* \in U \cap \chi$ such that $u(x^*) > u(x)$. This would contradict either $u(x^*) + v(\bar{p}) \leqslant 0$ which is impossible or imply $< x^*, \bar{p} > > 0$ which contradicts the feasibility of x^* and $\bar{p}$. The result is proved.

ON THE EXISTENCE OF JOINT PRODUCTION FUNCTIONS*

Rokaya Al-Ayat

Lawrence Livermore Laboratory
Livermore, California

Rolf Färe

Department of Economics
Southern Illinois University
Carbondale, Illinois

ABSTRACT

Within a general framework of production correspondences satisfying a set of weak axioms necessary and sufficient conditions for the existence of a joint production function are given. Without enforcing the strong disposability of inputs or outputs it is shown that a joint production function exists if and only if both input and output correspondences are strictly increasing along rays.

Joint production functions are frequently used in economics, however, it was not until Shephard in [6] defined such a notion within the general framework of production correspondences that its meaning became clear. The question of existence of these functions, dealt with in this paper, is yet to be settled. On this issue Shephard [8] wrote, "The joint production function is a tricky concept, seemingly simple but not shown to exist except under very restrictive conditions."

For a production technology with strongly disposable inputs and outputs Bol and Moeschlin [2], showed that continuity of both the input and the output correspondences together with essentiality of all inputs are sufficient for the existence of a joint production function. Later Bol in [1] showed that such a function would also exist if the essentiality condition is replaced by strict increasancy of the output correspondence in all inputs.

It is to be recalled that an output correspondence $x \rightarrow P(x) \in 2^{\mathbb{R}_+^m}$ is a mapping from input vectors $x \in \mathbb{R}_+^n$ into subsets $P(x) \in 2^{\mathbb{R}_+^m}$ of all output vectors obtainable by x. Inversely to $P(x)$ the input correspondence $u \rightarrow L(u) := \{x | u \in P(x)\}$ is the set of all input vectors x yielding at least an output vector u. In this paper the existence of a joint production function will be considered under the weak axioms as stated in [7]. Specifically neither the strong disposability of inputs or outputs (i.e., $x' \geqq x \in L(u) \Rightarrow x' \in L(u)$, $u' \leqq u \in P(x) \Rightarrow u' \in P(x)$ respectively) nor convexity of $P(x)$ or $L(u)$ are enforced. Having strong disposability of inputs means that if a subvector of inputs is kept constant while the remaining are increased,

*This research has been partially supported by the Office of Naval Research under Contract N00014-76-C-0134 with the University of California. Reproduction in whole or in part is permitted for any purpose of the United States Government.

output will never decrease implying there can be no congestion in the production system. In addition, strong disposability of outputs excludes their null jointness (see [9]); i.e., each output must be producible when others are not produced. Thus having only weak disposability of inputs (i.e., $P(\lambda \cdot x) \supset P(x), \lambda \geqslant 1$) and outputs (i.e., $L(\theta \cdot u) \subset L(u)$, $\theta \geqslant 1$) allow modelling of both congestion and null jointness.

As defined by Shephard [6], the joint production function relates input and output isoquants to each other. Recall that

$$ISOQ\ P(x) := \{u \mid u \in P(x),\ \theta \cdot u \notin P(x),\ \theta > 1\},\ P(x) \neq \{0\},$$

and

$$ISOQ\ L(u) := \{x \mid x \in L(u), \lambda \cdot x \notin L(u),\ \lambda < 1\},\ L(u) \not\ni \{0\},\ L(u) \neq \emptyset.$$

DEFINITION: The function $F: \mathbb{R}_+^m \times \mathbb{R}_+^n \to \mathbb{R}_+$ such that

(1) for $u^o \geqslant 0$, $ISOQ\ L(u^o) = \{x \mid F(u^o, x) = 0\}$, $L(u^o) \neq \emptyset$ and

(2) for $x^o \geqslant 0$, $ISOQ\ P(x^o) = \{u \mid F(u, x^o) = 0\}$, $P(x^o) \neq \{0\}$

is a joint production function.

An equivalent statement to the definition, to be used in the sequel, was proved by Bol and Moeschlin [2] namely:

LEMMA: A joint production function $F(u,x)$ exists if and only if for all $x \geqslant 0$[1], $P(x) \neq \{0\}$ and $u \geqslant 0$, $L(u) \neq \emptyset$, $u \in ISOQ\ P(x) \iff x \in ISOQ\ L(u)$.

THEOREM: For all $x \geqslant 0$, $u \geqslant 0$ such that $P(x) \neq \{0\}$, $L(u) \neq \emptyset$ with $x \to P(x)$ ($u \to L(u)$) satisfying the weak axioms, a necessary and sufficient condition for the existence of a joint production function $F(u,x)$ is

$$(*)\ ISOQ\ P(x) \cap ISOQ\ P(\lambda \cdot x) = ISOQ\ L(u) \cap ISOQ\ L(\theta \cdot u) = \emptyset$$

for all positive scalars λ, $\theta \neq 1$.

PROOF: To show the necessity of (*), assume there is a joint production function $F(u,x)$ and let $u \in ISOQ\ P(x) \cap ISOQ\ P(\lambda \cdot x)$. By the lemma, $x \in ISOQ\ L(u)$ and $\lambda \cdot x \in ISOQ\ L(u)$, $\lambda \neq 1$, which is a contradiction. Thus if a joint production exists, $ISOQ\ P(x) \cap ISOQ\ P(\lambda \cdot x)$ is empty for all positive scalers λ, $\lambda \neq 1$. A similar argument can be used to show that the existence of $F(u,x)$ implies that for all positive θ, $\theta \neq 1$, $ISOQ\ L(u) \cap ISOQ\ L(\theta \cdot u)$ is empty.

To show the sufficiency, assume that (*) holds, and that for $x \geqslant 0$, $P(x) \neq \{0\}$, $u \in ISOQ\ P(x)$ but $x \notin ISOQ\ L(u)$. From the definition of the isoquant, there exists a $\lambda < 1$ such that $\lambda \cdot x \in ISOQ\ L(u)$ implying that $u \in P(\lambda \cdot x)$. But from the weak disposability of inputs $P(\lambda \cdot x) \subset P(x)$ which together with (*) implies that $u \notin ISOQ\ P(x)$, a contradiction. Similarly it can be shown that having $ISOQ\ L(u) \cap ISOQ\ L(\theta \cdot u)$ empty would guarantee that $x \in ISOQ\ L(u) \nRightarrow u \in ISOQ\ P(x)$. Hence the sufficiency of (*) for the existence of a joint production function is proved. See lemma.

Q.E.D.

[1] $x \geqslant 0$ means $x \geqq 0$ but $x \neq 0$.

Continuity of the production correspondences has not been enforced. However, following an argument similar to that used by Bol and Moeschlin in [2] one can prove:

COROLLARY: If a joint production function exists, then both the input and the output correspondences are continuous along rays i.e., $P(\lambda^o \cdot x) = \overline{\bigcup_{0<\lambda<\lambda^o} P(\lambda \cdot x)}$ and $L(\theta^o \cdot u) = \overline{\bigcup_{\theta>\theta^o} L(\theta \cdot u)}$ respectively, with $u, x \neq 0$.

Note that continuity along rays together with strong disposability imply continuity (see [2] for definition).

Next, consider the production technology;

$$P(x_1, x_2) := \{\{(u_1, 0)\} \cup \{(0, u_2)\} | 0 \leqq u_i \leqq x_i, i = 1,2\}$$

and inversely

$$L(u_1, u_2) := \{\{(x_1, 0)\} \cup \{(0, x_2)\} | x_i \geqq u_i, i = 1,2\}.$$

The corresponding isoquants are given by

$$ISOQ\ L(u_1, u_2) = \{\{(x_1, 0)\} \cup \{(0, x_2)\} \mid x_i = u_i, i = 1,2\}$$

and

$$ISOQ\ P(x_1, x_2) = \{\{(u_1, 0)\} \cup \{(0, u_2)\} \mid u_i = x_i, i = 1,2\}.$$

In this example, the production correspondence satisfies the weak axioms, but neither strong disposability of inputs and outputs nor the essentiality condition (i.e., $P(x) \neq \{0\}$ implies $(x_1, x_2) > (0,0)$) used in [2] hold. Yet it is clear that a joint production function exists.

Finally, an example not satisfying the sufficiency conditions applied in [1] and [2] is given. Before introducing it the following proposition to be used, is proved.

PROPOSITION: If the production function $\phi(x) := \max \{u | x \in L(u)\}$, is continuous and strictly increasing along rays in the input space $\mathbb{R}^n_+$, $ISOQ\ L(u) = \{x | \phi(x) = u\}$, $u > 0$.

PROOF: Clearly $ISOQ\ L(u) \subset \{x | \phi(x) \geqq u\}$, $u > 0$; let $x^o \in \{x | \phi(x) > u\}$. Since ϕ is continuous along rays, $\{\lambda | \phi(\lambda \cdot x^o) > u\}$ is open implying that $x^o \notin ISOQ\ L(u)$, hence $ISOQ\ L(u) \subset \{x | \phi(x) = u\}$. Next assume $x^o \notin ISOQ\ L(u)$, $u > 0$, then since ϕ is strictly increasing along rays, if $x^o \in L(u)$, there is a $\lambda < 1$ such that $\phi(\lambda \cdot x^o) = u$ implying that $x^o \notin \{x | \phi(x) = u\}$.

Q.E.D.

Now, consider the output correspondence $x \rightarrow P(x) \subset [0, +\infty)$,

$$P(x) := \{u \in R_+ : u \leqq \phi(x)\}$$

$$\phi(x) := \begin{cases} A \cdot [(1-\delta)(x_1 - \gamma x_2)^{-\rho} + \delta x_2^{-\rho}]^{-1/\rho} \\ \text{for } (x_1 - \gamma x_2) \geqq 0 \\ 0 \text{ otherwise} \end{cases}$$

where the parameters of the WDI — production function $\phi(x)$ are $A > 0$, $\delta \in (0,1)$, $\gamma \in (0, \infty)$ and $\rho \in (-1, 0)$ (see [3]). For these values of the parameters, $\phi(x)$ is upper semi-continuous which is equivalent to $P(x)$ being upper hemi-continuous (see [5], p. 22) also

$x_2 = 0$ does not imply $P(x) = \{0\}$ and ϕ is not increasing in x_2. Thus $P(x)$ does not meet the continuity requirement of [1] and [2] nor does it meet the other sufficiency condition of [2] (essentiality of all factors) or [1] (strict increasancy in all factors).

Using the proposition above the isoquants of $P(x)$ and $L(u)$ are easily computed to be,

$$ISOQ\ P(x) = \{u | u = \phi(x)\} \text{ and } ISOQ\ L(u) = \{x | \phi(x) = u\}.$$

Thus, $x \in ISOQ\ L(u) \Leftrightarrow u \in ISOQ\ P(x)$, showing that under the weak axioms for a production technology, the sufficient conditions found in [1] and [2] need not hold for a joint production function to exist.

ACKNOWLEDGMENT

The authors sincerely thank Professor Ronald W. Shephard for his suggestions and helpful comments.

BIBLIOGRAPHY

[1] Bol, G., "Produktionskorrespondenzen und Existenz Skalarwertiger Produktionsfunktionen bei der Mehrgüterproduktion," Karlsruhe, (1976).

[2] Bol, G. and O. Moeschlin, "Isoquants of Continuous Production Correspondences," Naval Research Logistics Quarterly, Vol. 22, pp. 391-398, (1975).

[3] Färe, R. and L. Jansson, "On VES and WDI Production Functions," International Economic Review, Vol. 16, pp. 745-750, (1975).

[4] Färe, R. and L. Jansson, "Joint Inputs and the Law of Diminishing Returns," Zeitschrift für Nationalökonomie, Vol. 36, pp. 407-416, (1976).

[5] Hildenbrand, W., *Core and Equilibra of a Large Economy,* (Princeton University Press, 1974).

[6] Shephard, R.W., *Theory of Cost and Production Functions,* (Princeton University Press, 1970).

[7] Shephard, R.W., "Semi-Homogeneous Production Functions," Lecture Notes in Economics and Mathematical Systems, Volume 99, Production Theory, Berlin, Springer-Verlag, (1974).

[8] Shephard, R.W., "On Household Production Theory," ORC 76-24, Operations Research Center, University of California, Berkeley, (1976).

[9] Shephard, R.W. and R. Färe, "The Law of Diminishing Returns," Zeitschrift für Nationalökonomie, Vol. 34, pp. 69-90, (1974).

THE PURE FIXED CHARGE TRANSPORTATION PROBLEM

John Fisk

School of Business
State University of New York at Albany
Albany, New York

Patrick McKeown

College of Business Administration
University of Georgia
Athens, Georgia

ABSTRACT

The pure fixed charge transportation problem (PFCTP) is a variation of the fixed charge transportation problem (FCTP) in which there are only fixed costs to be incurred when a route is opened. We present in this paper a direct search procedure using the LIFO decision rule for branching. This procedure is enhanced by the use of 0-1 knapsack problems which determine bounds on partial solutions. Computational results are presented and discussed.

. INTRODUCTION

The pure fixed charge transportation problem (PFCTP) deals with the optimal allocation f supply S_i available at source $i = 1, 2, \ldots, m$ in order to meet demand D_j at destination $= 1, 2, \ldots, n$. Before any goods can be shipped from i to j a fixed charge, f_{ij}, must be paid. he objective is then to minimize the total cost of shipping available goods to meet the equired demands.

One would wish to solve the pure fixed charge transportation problem in those situations here the cost to transport goods over an arc must be paid as a lump sum rather than as per nit costs. An example would be leasing trucks to move goods between supply points and emand points. So long as the amount demanded is less than the capacity to the truck for that oute, the cost of moving any amount of goods greater than zero is approximately the same, e., the leasing cost, fuel, the driver's salary are fixed. In this case, we would wish to deter- ine the set of routes which would allow us to satisfy total demand at a minimum possible sum f these lump or fixed costs.

Mathematically the problem may be formulated as follows:

1) Minimize $$Z = \sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij} y_{ij}$$

2) subject to $$\sum_{j=1}^{n} x_{ij} = S_i \qquad i = 1, 2, \ldots, m,$$

$$\sum_{i=1}^{m} x_{ij} = D_j \qquad j = 1, 2, \ldots, n, \tag{3}$$

$$x_{ij} \geqslant 0, \ \forall_{i,j} \tag{4}$$

$$y_{ij} = \begin{cases} 1 \text{ if } x_{ij} > 0 \\ 0 \text{ otherwise} \end{cases} \tag{5}$$

We are assuming that $\sum_{i=1}^{m} S_i = \sum_{j=1}^{n} D_j$ and that the f_{ij}'s are integer.

In this paper, we will present a direct search procedure for solving the PFCTP which utilizes the unique structure of the constraints, (2) and (3), to derive bounds that lead to an efficient search over the possible solutions. In Section II, we will discuss solution procedures for a closely related problem, the fixed charge transportation problem, as they relate to the PFCTP and suggest a procedure for solving the PFCTP. Section III will discuss development of bounds and presents the iterative procedure. Computational results are presented and discussed in Section IV.

II. SOLUTION PROCEDURES FOR THE FIXED CHARGE TRANSPORTATION PROBLEM (FCTP)

Discussion of the PFCTP in the literature is limited. Numerous techniques for solving the closely related fixed charge transportation problem (FCTP) are presently available, however. The FCTP can be specified mathematically as follows:

$$\text{Minimize} \qquad Z = \sum_{i=1}^{m} \sum_{j=1}^{n} (f_{ij} y_{ij} + c_{ij} x_{ij}) \tag{6}$$

$$\text{subject to} \qquad (2), (3), (4), \text{ and } (5).$$

We also assume that $\sum_{i=1}^{m} S_i = \sum_{j=1}^{n} D_j$ and that the fixed costs, f_{ij}'s are integer. In this case, we also have the usual per unit transportation costs c_{ij}.

Research into solving the FCTP may be classified as either heuristic or exact (algorithmic). We will be interested in the latter. Some of these are Murty [15], Gray [6], Kennington and Unger [9, 10], McKeown [14], Kennington [10, 11], Barr [2], Frank [5], and Steinberg [17]. With the exception of extreme point ranking procedures such as those presented by Murty and by McKeown, any of the above procedures could be applied to solving PFCTP. Most of these would not be expected to be efficient, either because they are designed to solve more general types of fixed charge problems and do not take full advantage of the special constraints (2) and (3), or because they have been shown to be efficient only when variable costs dominate fixed costs. Examples of heuristic methods are [1] and [13].

Two procedures which would appear to be useful in solving PFCTP, however, are those of Gray and Kennington. In Gray's procedure, a series of tests are developed which enable him to decrease the number of vertices for which he must find the corresponding feasible solution in order to find a satisfactory assignment of routes given specific values of the logic variables

y_{ij}. Kennington introduces a branch-and-bound procedure for solving FCTP which employs a relaxation corresponding to the following Hitchcock Transportation Problem (TP):

(7) Minimize $\sum_{i=1}^{n}\sum_{j=1}^{n}[(f_{ij}/u_{ij}) + c_{ij}]x_{ij}$

subject to (2), (3), and (4)

and where $u_{ij} = \min(S_i, D_j)$. The formulation for TP above was introduced by Balinski [1] in an approximate procedure for solving FCTP. Solving TP at each node of his branch-and-bound tree, Kennington is able to calculate effective bounds and to determine simple penalties and feasibilities useful in directing the search procedure. His methodology could be applied to solving PFCTP by simply setting $c_{ij} = 0$, all for i, j in (7).

III. DIRECT SEARCH PROCEDURE FOR SOLVING THE PFCTP

Using the terminology of Geoffrion and Marsten [5] we solve the problem PFCTP by separating its set of feasible solutions into subproblems called candidate problems (CP) by assigning values to a subset of the variables (y_{ij}). A particular (CP) is fully defined by specifying the elements contained in each of two sets J_0 and J_1, which represent the set of transportation routes assigned "closed" (i.e., $y_{ij} = 0$) and "open" (i.e., $y_{ij} = 1$), respectively. The remaining elements reside in the set J_2 and are referred to as "free." The sets J_0, J_1, and J_2 are mutually exclusive and collectively exhaustive.

Our enumerative scheme is similar in most respects to a more traditional branch-and-bound scheme employing a direct search (single branch) strategy. In constructing our branching tree, we proceed through successive levels of the tree by choosing a route from the set J_2 and assigning it to J_1. We assign this route to J_0 in the branching tree only upon backtracking. A strict LIFO (last-in, first-out) backtracking rule is observed.

Three factors critical to the computational efficiency of the above approach are (1) the effort required to solve for the bound associated with a given candidate problem, and the quality of the bound produced, (2) the specification of rules useful in identifying (CP's) which cannot be optimal, and (3) the choice of a separation variable from among those in J_2. Sections IIIa and IIIb detail two bounding procedures, the row feasibility test and the column feasibility test. These tests are easily applied and, when used in conjunction with one another, can yield efficient bounds. Section IIIc specifies a test attributable to Hirsch and Dantzig [8] which serves to limit the number of routes assignable to set J_1. Section IIId outlines the rules used for selecting a separation variable from J_2. In the discussion that follows, we introduce a set $J_s = \{J_0 + J_1\}$ which we refer to as a partial solution to PFCTP.

IIIa. Row Feasibility Test

We define the cost for row feasibility for row i, RF_i, in terms of the least cost set of demanders necessary to absorb the supply S_i given the partial solution J_s. We further define AD_i as the total demand assigned to row i given the partial solution J_s, where

(8) $$AD_i = \sum_j D_j y_{ij}$$

and all free variables in row i are assumed closed. If $AD_i \geqslant S_i$, then RF_i is simply the total cost of the open routes in row i, i.e.,

(9) $$RF_i = \sum_j f_{ij} y_{ij}.$$

If $AD_i < S_i$, however, we must determine the minimum additional cost which must be incurred in order to satisfy a necessary condition for row feasibility. To do so, we solve the following 0-1 knapsack problem relative to the set of unassigned routes from supply i:

$$\text{(10)} \qquad \text{Minimize} \qquad \Pi_i = \sum_{j \in J_2\{i\}} f_{ij} u_{ij}$$

$$\text{(11)} \qquad \text{subject to} \qquad \sum_{j \in J_2\{i\}} D_j u_{ij} \geqslant d_i$$

$$\text{(12)} \qquad u_{ij} = 0,\ 1 \qquad \forall_j \in J_2\{i\}$$

where $d_i = S_i - AD_i$ and $J_2\{i\}$ = the set of unassigned routes from supply i. Then given that assigned demand is less than the available supply for row i, the minimum cost necessary to obtain row feasibility becomes

$$\text{(13)} \qquad RF_i = \sum_{j \in J_s} f_{ij} y_{ij} + \Pi_i.$$

The applicability of the knapsack relation for determining RF_i is based upon the ability to solve problems such as (10) - (12) with minimal effort [3], [7], [18]. Such relations can yield efficient bounds and have been successfully applied in solving the generalized assignment problem [16] and in solving warehouse location problems [12].

Since the row feasibility test described above can be applied for each row (supplier) i given the partial solution J_s, the value $RF = \sum_i RF_i$ becomes a lower bound on the sum of fixed charges required for a feasible completion to J_s. If this value of RF is greater than or equal to Z (the current best known feasible solution), we have fathomed J_s.

As pointed out previously, the knapsack relation which we employ for determining the bound RF requires relatively little computational effort. Even so, the computational efficiency of our procedure increases as the number of such knapsack problems necessary to solve PFCTP decreases. The paragraph that follows indicates the procedures we employ in order to reduce the computational cost of using our knapsack relation.

At the initialization of our procedure—when all routes are considered free—we calculate RF by applying (10) - (12) for each row i as previously described, then store the knapsack solution for each such row. Thereafter, as we assign a route to be open or closed, we update the bound RF by adjusting the knapsack solution and its objective value for the corresponding row only. Also, upon assigning a route to be open ($y_{ij} = 1$) the knapsack relation need be applied only if:

(1) $AD_i < S_i$ and

(2) the route (i, j) is not one of the assigned open routes in the stored knapsack solution for its row i.

Similarly, upon assigning a route to be closed ($y_{ij} = 0$) the knapsack relation need be applied only if:

(1) $AD_i < S_i$ and

(2) the route (i, j) is not one of the assigned closed routes in the stored knapsack solution for its row i.

IIIb. Column Feasibility Test

In determining column feasibility for column j, CF_j, we use procedures strictly analogous to those described for determining the cost for row feasibility. We define AS_j as the total supply assigned to column j given the partial solution J_s, where

$$AS_j = \sum_i S_i y_{ij} \tag{14}$$

and all free variables in column j are assumed closed. If $AS_j \geqslant D_j$, then CF_j is simply the total cost of the open routes in column j; i.e.,

$$CF_j = \sum_i f_{ij} y_{ij}. \tag{15}$$

If $AS_j < D_j$, however, we must determine the minimum additional cost which must be incurred in order to satisfy a necessary condition for column feasibility. To do so, we solve the following 0-1 knapsack problem relative to the free variables in column j:

$$\text{Minimize} \quad \Pi_j = \sum_{i \in J_2(j)} f_{ij} v_{ij} \tag{16}$$

$$\text{subject to} \quad \sum_{i \in J_2(j)} S_i v_{ij} \geqslant d_j \tag{17}$$

$$v_{ij} = 0, 1 \qquad \forall_{i \in J_2(j)} \tag{18}$$

where $d_j = D_j - AS_j$ and $J_{2(j)}$ = the set of unassigned routes to demand j. Given that assigned supply is less than the necessary demand for column j, the minimum cost necessary to obtain column feasibility becomes

$$CF_j = \sum_{i \in J_s} f_{ij} y_{ij} + \Pi_i. \tag{19}$$

The rules for applying the knapsack relation (16) - (18) follow closely those defined for rows in the preceding subsection. Also, since this column feasibility test described above can be applied for each column (demander) j given the partial solution J_s, the value $CF = \sum_j CF_j$ becomes a lower bound on the sum of fixed charges required for a feasible completion to J_s. The best available bound assignable to the partial solution J_s becomes max (RF, CF).

IIIc. Basis Constraint Test

Hirsch and Dantzig showed that, for any fixed charge problem, an optimal solution occurs as an extreme point of the (continuous) constraint set. This implies that the x_{ij} values corresponding to a partial solution of the y_{ij}'s must be linearly independent and must not be infeasible. Any partial solution which does not satisfy these conditions may be terminated. Also, the maximum number of nonzero elements (i.e., routes (i, j) for which $y_{ij} = 1$) in a basic solution is $m + n - 1$.

IIId. Choice of Separation Variable

The separation variable $y_{i \bullet j \bullet}$ is chosen from amongst the sets of variables $u^* = u_{ij}^*$ and $v^* = v_{ij}^*$. We define u^* as the optimal set of open variables obtained by solving (10) - (12) for each row i for which $AD_i < S_i$ given J_s, and v^* as the optimal set of open variables obtained by solving (16) - (18) for each column j for which $AS_j < D_j$ given J_s. If $\Pi_{i(j)}$ represents the objective value of (10) - (12) given the closure of route (i, j) in row i, then $p_{ij} = \Pi_{i(j)} - \Pi_i$ is

the penalty associated with the closure of route (i, j). Similarly, $\Pi_{j(i)}$ represents the objective value of (16) - (18) given the closure of route (i, j) in column j, and $q_{ij} = \Pi_{j(i)} - \Pi_j$ is the penalty associated with the closure of route (i, j). A nonzero penalty need be obtained only for those routes included in u^* and v*.

Given the determination of the set of penalties $p = p_{ij}$ associated with the closure of each route in u^*, the maximum increase in the value of row feasibility RF given the closure of any route in u^* becomes $p_m = \max_{(i,j) \in u^*} (p_{ij})$. Similarly, the maximum increase in the value for column feasibility CF given the closure of any route in v* becomes $q_m = \max_{(i,j) \in v^*} (q_{ij})$. The separation variable $y_{i \bullet j \bullet}$ is therefore that currently unassigned variable whose closure would yield the greatest bound associated with J_s:

(20).
$$r_{i \bullet j \bullet} = \max\,(RF + p_m,\ CF + q_m).$$

In the event that u^* and v* are empty sets (i.e., $AD_i \geqslant S_i$, $\forall$ and $AS_j \geqslant D_j$, $\forall_j$), then the separation variable $y_{i \bullet j \bullet}$ becomes the currently assigned variable having minimum fixed cost.

This completes the discussion of the iterative procedure used, and the set of tests employed in order to eliminate partial solutions. For a simple example which illustrates the application of these tests within the procedure, the reader is referred to the Appendix.

IV. COMPUTATIONAL EXPERIENCE

The algorithm as described here, PURFIX, has been programmed in FORTRAN IV and run on the CYBER 70/74 using the time-share mode. A series of 5 × 5 problems similar to those originally tested by Kennington [11] were run. These problems had uniformly generated supplies and demands over the range 1-999 with uniformly generated costs using various ranges. The cost parameters and test results are shown in Table 1 below. In addition, the Kennington code was obtained for use as a benchmark to determine the relative efficiency of our algorithm. These results are also shown in Table 1. All solution times are an average of five problems. Also shown for both procedures is the difference between the fastest and the slowest solution times for each set of problems (the range).

TABLE 1

Problem Set	Fixed Cost Range	Average PURFIX Time	Range	Average Kennington Time	Range
1	257 - 457	5.964	12.936	10.897	39.859
2	614 - 814	12.884	21.604	32.285	122.542
3	1328 - 1528	8.069	15.401	33.665	81.796
4	1231 - 3231	2.784	2.577	4.626	5.507
5	3463 - 5463	5.481	6.212	8.017	12.368
6	34700 - 36700	13.239	26.422	7.645	17.143
7	66400 - 76400	8.042	15.327	34.021	81.823
8	2570000 - 4570000	4.427	9.393	11.532	42.555

All times are in CPU seconds and do not include problem generation.

As may be seen in Table 1, PURFIX is faster than the Kennington code in all cases except one. This case happens to be where the fixed cost range is fairly small compared to the magnitude of the fixed costs. Under these conditions PURFIX would be expected to have difficulty distinguishing the optimal solution. In six of the remaining seven cases, PURFIX is at least twice as fast. It can also be noted that for both procedures the range values are fairly large. This implies that the effectiveness of either procedure for pure fixed charge transportation problems is highly dependent upon the particular problem being solved and can vary greatly from problem to problem. As with the solution times, the range values are less for PURFIX in seven out of the eight cases.

To test the effectiveness of the PURFIX procedure relative to problem size, we ran six sets of three problems each. All sets were similar except for problem size. The fixed charges were randomly generated with values between 0 and 10 and demands were generated with values between 10 and 100 in increments of 10. The supplies were generated in a similar manner in such a way that total supplies equal total demands. For each problem set, there were five supplies but differing numbers of demands. The results from the computational testing is shown in Table 2 with average times (CPU seconds) and ranges being shown for each problem set.

TABLE 2

Problem Set	Size ($m \times n$)	Average Solution Time	Range
1	5 × 5	.196	.248
2	5 × 7	.467	1.171
3	5 × 9	2.250	4.045
4	5 × 10	.377	.362
5	5 × 13	2.451	2.963
6	5 × 15	***	***

In Table 2 we see that while the number of arcs has a definite effect on solution time, it is not always the only determinant of difficulty of solution. This is evident from the fact that problem set four with 50 arcs was solved in less time than that required for problem sets two and three, each having fewer arcs. PURFIX was unable to solve any problems having 75 or more arcs in less than an average of 50 seconds.

Another factor that could effect ease of solution is the shape of the problem. By this is meant the relationship between the number of supplies and number of demands. The problems tested in Table 2 with the exception of Set 1 were all rectangular problems with more demands than supplies. In Table 3 we have also tested "square" problems, i.e., those with equal numbers of supplies and demands. All characteristics other than shape were the same as for Table 2.

TABLE 3

Problem Set	Size ($m \times n$)	Average Solution Time	Range
A	6 × 6	.597	1.230
B	7 × 7	2.451	3.489
C	8 × 8	1.873	1.895

If we compare the problem sets in Table 3 to problem sets in Table 2 having approximately the same number of arcs, i.e., problem sets 2, 4, and 5, we can get some idea of the effect of shape on ease of solution. However, the comparisons do not show any clear difference in solution times that could be attributed to the shape of the problem.

In summary, these computational results imply that while the size of the problem has a definite effect, it appears that ease of solution is highly dependent on some combination of costs and supplies and demands. The exact effect is unclear but definitely deserves further research.

REFERENCES

[1] Balinski, M.L., "Fixed Cost Transportation Problem," Naval Research Logistics Quarterly, Vol. 8, pp. 41-54 (1961).

[2] Barr, R.L., "The Fixed Charge Transportation Problem," presented at the joint National Meeting of ORSA and TIMS in Puerto Rico (1975).

[3] Fisk, J., "An Initial Bounding Procedure for use with 0-1 Single Knapsack Algorithms," Opsearch, Vol. 14, pp. 88-98 (1977).

[4] Frank, R., "On the Fixed Charge Hitchcock Transportation Problems," (dissertation), Johns Hopkins (1972).

[5] Geoffrion, A.M. and R.E. Marsten, "Integer Programming Algorithms: A Framework and State-of-the-Art Survey," in *Perspectives on Optimization*, Geoffrion, Ed., Addison-Wesley (1972).

[6] Gray, P., "Exact Solution of the Fixed-Charge Transportation Problem," Operations Research, Vol. 19, pp. 1529-38 (1971).

[7] Greenberg, H. and R. Hegerich, "A Branch Search Algorithm for the Knapsack Problem," Management Science, Vol. 16, pp. 327-32 (1970).

[8] Hirsch, W.M. and G.B. Dantzig, "The Fixed Charge Problem," Naval Research Logistics Quarterly, Vol. 15, pp. 413-24 (1968).

[9] Kennington, J. and V. Unger, "The Group-Theoretic Structure in the Fixed-Charge Transportation Problem," Operations Research, 21, pp. 1142-1153 (1973).

[10] Kennington, J.L. and V.E. Unger, "A new Branch and Bound Algorithm for the Fixed-Charge Transportation Problem," Management Science, Vol. 22, pp. 1116-1126 (1976).

[11] Kennington, J.L., "The Fixed-Charge Transportation Problem: A Computational Study with a Branch- and -Bound Code," AIIE Transactions, Vol. 7, pp. 241-247 (1975).

[12] Khumawala, B.M. and U. Akinc, "An Efficient Branch and Bound Algorithm for the Capacitated Warehouse Location Problem," Management Science, Vol. 23, pp. 585-594 (1977).

[13] Kuhn, H.W. and W.J. Baumol, "An Approximative Algorithm for the Fixed-Charges Transportation Problem," Naval Research Logistics Quarterly, Vol. 9, pp. 1-16 (1962).

[14] McKeown, P.G., "A Vertex Ranking Procedure for the Linear Fixed Charge Problem," Operations Research, Vol. 23, No. 6, pp. 1183-1191 (1975).

[15] Murty, K.G., "Solving the Fixed Charges Problem by Ranking the Extreme Points," Operations Research, Vol. 16, pp. 268-79 (1968).

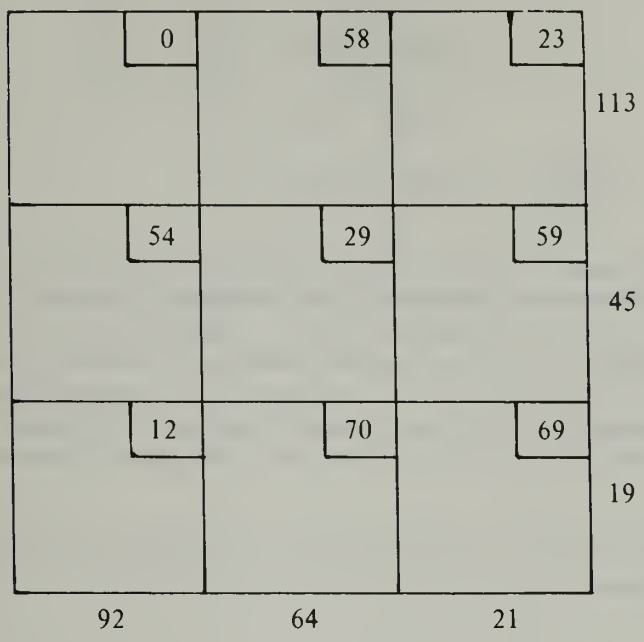
[16] Ross, G.T. and R.M. Soland, "A Branch and Bound Algorithm for the Generalized Assignment Problem," Mathematical Programming, 8, 91-103 (1975).

[17] Steinberg, D., "The Fixed Charge Problem," Naval Research Logistics Quarterly, 17, No. 2, pp. 217-234 (1970).

[18] Zoltners, A.A., "A Direct Descent Binary Knapsack Algorithm," working paper #75-31, University of Massachusetts (1975).

APPENDIX

For purposes of illustration, consider the following simple example problem:



Calculation of row feasibility for the first row, RF_1, requires solution of the following single knapsack problem:

$$\begin{aligned} \text{Minimize} \quad & \Pi_1 = 0u_{11} + 58u_{12} + 23u_{13} \\ \text{Subject to} \quad & 92u_{11} + 64u_{12} + 21u_{13} \geqslant 113 \\ & u_{ij} = 0,\ 1,\ \forall_j \end{aligned}$$

The optimal solution to the above problem is $u^* = (u_{11} = 1,\ u_{13} = 1,\ u_{12} = 0)$ and $\Pi_1 = 23$. Additional knapsack solutions can be obtained for rows 2 and 3 so that the following row feasibility table can be constructed:

i	Π_i			
1	23	1 / ∞		1 / 35
2	29		1 / 25	
3	12	1 / 57		

Each cell in the table above can be interpreted as follows: a "1" in the upper diagonal indicates that the corresponding transportation route is assigned to be "open" in the optimal knapsack solution, while the value in the lower diagonal indicates the penalty associated with closing that route. Empty cells indicate that the corresponding transportation route is "closed" in the knapsack solution for its row. Row feasibility is $RF = \sum_i \Pi_i = 23 + 29 + 12 = 64$. A table similar to that for rows can be constructed for columns as follows:

j	Π_j			
1	0	1 / ∞		
2	58	1 / 41		
3	23	1 / 36		

Column feasibility, $CF = \sum_j \Pi_j = 0 + 58 + 23 = 81$.

Since an infinite penalty is associated with the closure of route (1, 1), then y_{11} is set to 1 and assigned to J_s. Since column feasibility is now obtained for column 1, the next variable is chosen from rows 1, 2, and 3 and columns 2 and 3. Since $p_m = p_{31} = 57$ and $q_m = q_{12} = 41$, the next variable assigned to J_s is that currently unassigned variable $y_{i \cdot j \cdot}$ for which $r_{i \cdot j \cdot} = \max (64 + 57, 81 + 41) = 122$ and $y_{i \cdot j \cdot} = y_{12}$. Assigning y_{12} to one and adding it to J_s simultaneously satisfies row feasibility in row 1 and column feasibility in column 2. Row feasibility is increased from 64 to 99, since route (1, 2) was not in the optimal knapsack solution for calculating RF_1.

The procedure continues in a similar manner. The solution tree for our example problem is found in Figure 1 below:

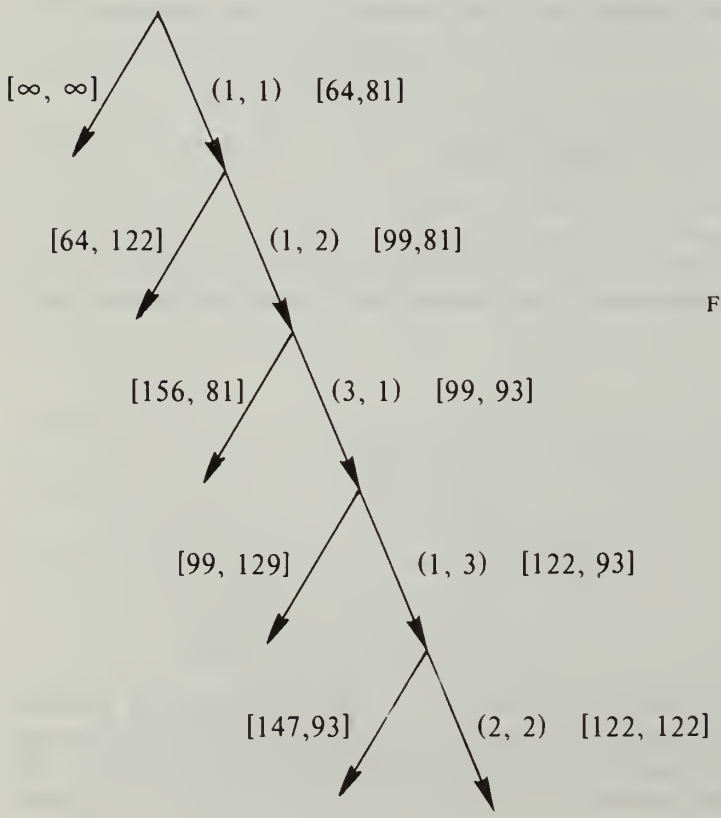


FIGURE 1. Branching tree for example problem

Note that the route assigned at each level of the tree is in parentheses, while the values for row and column feasibility are in brackets. The unit flows associated with the solution obtained in Figure 1 are as follows:

73	19	21	113
	45		45
19			19
92	64	21	

Optimal solution value is 122.

A HEURISTIC ROUTINE FOR SOLVING LARGE LOADING PROBLEMS

John C. Fisk

State University of New York at Albany
Albany, New York

Ming S. Hung

Cleveland State University
Cleveland, Ohio

ABSTRACT

The loading problem involves the optimal allocation of n objects, each having a specified weight and value, to m boxes, each of specified capacity. While special cases of these problems can be solved with relative ease, the general problem having variable item weights and box sizes can become very difficult to solve. This paper presents a heuristic procedure for solving large loading problems of the more general type. The procedure uses a surrogate procedure for reducing the original problem to a simpler knapsack problem, the solution of which is then employed in searching for feasible solutions to the original problem. The procedure is easy to apply, and is capable of identifying optimal solutions if they are found.

. INTRODUCTION

The loading problem involves the optimal allocation of n objects $i = 1, 2, \ldots, n$, each having a given value c_i and weight w_i, to m boxes, $j = 1, 2, \ldots, m$, each having capacity b_j. Several types of loading problems exist, as indicated in Eilon and Christofides [4]. Two of these are:

PROBLEM 1 (*P*1): Given that $\sum_j b_j \geqslant \sum_i w_i$ determine the minimum number of boxes required to accommodate all items.

PROBLEM 2 (*P*2): Given that $\sum_j b_j < \sum_i w_i$ (or $\sum_j b_j \geqslant \sum_i w_i$ but not all objects can be accommodated), determine the maximum value of objects accommodated in the boxes.

The integer program for problem (P1) can be written as follows:

1) (P1) minimize $\sum_j d_j y_j$

2) subject to $\sum_j x_{ij} = 1, \quad \forall i$

$$\sum_i w_i x_{ij} \leqslant b_j y_j, \ \forall j \tag{3}$$

$$y_i = 0, 1 \ \forall i, \ x_{ij} = 0, 1, \ \forall i, j \tag{4}$$

where

$$y_j = \begin{cases} 1 & \text{if box } j \text{ contains one or more objects} \\ 0 & \text{otherwise} \end{cases}$$

$$x_{ij} = \begin{cases} 1 & \text{if object } i \text{ is placed in box } j \\ 0 & \text{otherwise} \end{cases}$$

and d_j is the cost of box j. For $d_j = 1, \ \forall j$ the problem reduces to determining the minimum number of boxes required to hold all objects. If $d_j = b_j, \ \forall j$, the above problem becomes that of determining the minimum capacity set of boxes required to hold all objects.

For (P2) the integer programming problem is

$$\text{(P2)} \quad \text{maximize} \sum_i c_i x_{ij} \tag{5}$$

$$\text{subject to} \sum_j x_{ij} \leqslant 1, \forall i \tag{6}$$

$$\sum_i w_i x_{ij} \leqslant b_j, \forall j \tag{7}$$

$$x_{ij} = 0, 1, \forall i, j \tag{8}$$

Eilon and Christofides present a heuristic procedure for solving a special case of problem (P1) in which $d_j = 1, \ \forall_j$. In addition, they introduce an enumerative algorithm based on the work of Balas [2] which yields satisfactory results. A more efficient algorithmic procedure for this problem which again takes advantage of uniform box costs is presented by Hung and Brown [10].

For solving problem (P2), Ingargiola and Korsh [12] introduce an ordering relation which allows a reduction in the amount of searching required within an enumerative scheme. Hung and Fisk [11] present procedures which rely on Lagrangian and surrogate relaxations to yield good bounds in a branch and bound scheme. Similar procedures have been developed by Martello and Toth [13]. Each of these procedures appears to yield satisfactory results as long as the number of items is small ($\leqslant 100$) and the number of boxes does not exceed three.

This paper presents a simple and effective heuristic procedure for solving loading problems (P1) and (P2) of much larger scale than those that have been attempted before. The procedure is similar to that of Glover [9] in that it uses surrogate constraints to obtain some feasible solutions, but it has two distinctive features usually not found in heuristic procedures. One is that our procedure uses the surrogate constraints to reduce the problems (P1) and (P2) to simpler problems which in fact are the well known knapsack problems. Then we use the solutions to the knapsack problems to reduce the set of variables to be considered later on. Another feature is that our procedure will identify optimal solutions if they are found. More specifically, if the reduced set of variables produces a feasible solution then we know that the solution is optimal.

II. SURROGATE RELAXATIONS FOR (P1) AND (P2)

The concept and applicability of surrogate relaxation were introduced by Glover [7], [8] while useful refinements were suggested by Balas [3] and Geoffrion [6]. Surrogate relaxation in its simplest form is to replace a set of constraints by a single constraint (the surrogate constraint). For example, for problem (P1) a nonnegative vector of real numbers $\alpha = (\alpha_j)$ can be used to aggregate the n constraints in (3) into a single one,

$$\sum_j \sum_i \alpha_j \, w_i \, x_{ij} \leqslant \sum_j \alpha_j \, b_j \, y_j \tag{9}$$

Similarly for (P2), a nonnegative real vector $\pi = (\pi_j)$ can be used to combine the n constraints in (7) into the following,

$$\sum_i \sum_j \pi_j \, w_i \, x_{ij} \leqslant \sum_j \pi_j \, b_j \tag{10}$$

Let $(P1_\alpha)$ and $(P2_\pi)$ respectively denote the surrogate relaxations of (P1) and (P2). The relaxations $(P1_\alpha)$ and $(P2_\pi)$ can provide good lower bounds to the optimal solution for their respective original problems given a suitable choice of multipliers. Balas [3], Geoffrion [6] and Hung and Fisk [11] have shown that one suitable choice is to set them equal to the optimal dual multipliers of the aggregated constraints of the linear programming problem. For example, let $\bar{\pi} = (\bar{\pi}_j)$ represent the set of optimal dual multipliers of constraints (6) in the linear program of (P2). The linear program is obtained by replacing the 0-1 constraints (8) with unit intervals $0 \leqslant x_{ij} \leqslant 1$ for all i, j. Hung and Fisk [11] showed that

$$\bar{\pi}_j = c_t / w_t \text{ for all } j$$

where t is the smallest object index such that

$$\sum_{i \leqslant t} w_i \geqslant \sum_j b_j$$

Items are assumed to be ordered such that

$$c_1/w_1 \geqslant c_2/w_2 \geqslant \ldots \geqslant c_n/w_n.$$

For (P1), if $\bar{\alpha} = (\bar{\alpha}_j)$ represents the set of optimal dual multipliers of constraints (2) in the linear program of (P1), then:

$$\bar{\alpha} = d_u / b_u \text{ for all } j$$

where u is the smallest box index such that

$$\sum_{j \leqslant u} b_j \geqslant \sum_i w_i$$

The boxes are assumed ordered such that

$$d_1/b_1 \leqslant d_2/b_2 \leqslant \ldots \leqslant d_u/b_u \leqslant \ldots \leqslant d_n/b_n$$

Since $\bar{\alpha}_j$ is a constant for all j, constraint (9) can be simplified and the surrogate problem $(P1_{\bar{\alpha}})$ becomes a single knapsack problem as follows:

$$(P1_{\bar{\alpha}}) \quad \text{minimize} \sum_j d_j \, y_j$$

$$\text{subject to} \sum_i w_i \, x_i = \sum_i w_i \leqslant \sum_j b_j \, y_j$$

$$y_j = 0, \, 1 \; \forall j$$

where:

$$x_i = \sum_j x_{ij} = 1, \quad \forall i.$$

Similarly, $(P2_{\bar{\pi}})$ has the following simple form:

$$(P2_{\bar{\pi}}) \quad \text{maximize} \sum_i c_i x_i$$

$$\text{subject to} \sum_i w_i x_i \leqslant \sum_j b_j$$

$$x_i = 0, 1, \forall i$$

where:

$$x_i = \sum_j x_{ij}.$$

The single knapsack problems defined within $(P1_{\bar{\alpha}})$ and $(P2_{\bar{\pi}})$ can be easily solved (see e.g., Ahrens and Finke [1], Fisk [5]). It is clear that an optimal solution to the relaxed problem, either $(P1_{\bar{\alpha}})$ or $(P2_{\bar{\alpha}})$, may violate the original constraints — (3) in (P1) and (7) in (P2) — because the assignment of items to boxes is ignored in the relaxed problems. However, if a feasible assignment can be found among the items and boxes identified in the optimal solutions to the relaxed problems, then optimal solutions to the original problems are found. Let S represent a set of items while T represents a set of boxes. For (P1) let

$S \equiv$ {all items}

$T \equiv$ {all boxes for which $y_j = 1$ in the optimal solution to $(P1_{\bar{\alpha}})$}

while for (P2) let

$S \equiv$ {all items for which $x_i = 1$ in the optimal solution to $(P2_{\bar{\pi}})$}

$T \equiv$ {all boxes}

When solving (P1), if an assignment of S to T is found which satisfies constraint sets (2) - (4) then this assignment represents a feasible and optimal solution to (P1). Similarly, when solving (P2), if an assignment of S to T is found which satisfies constraint sets (6) - (8) then this assignment represents a feasible and optimal solution to (P2). The following section describes a procedure which searches for feasible and (therefore) optimal assignments of S to T.

III. AN EXCHANGE ROUTINE FOR (P1) AND (P2)

The exchange routine to be described is similar in many respects to the heuristic algorithm presented by Eilon and Christofides. For a particular problem, the procedure uses only the sets S and T as previously defined, and searches for an assignment of items which is feasible for the original problem. If such a feasible assignment is found, then it also represents an optimal assignment to the original problem.

The exchange procedure we use is the following:

STEP 1: Solve $(P1_{\bar{\alpha}})$ [or $(P2_{\bar{\pi}})$ if the problem is to solve (P2)] and identify the set of items S and set of boxes T. Place the set of items in S in a list in descending order according to weight. The set of boxes in T can be placed in a new list in any order.

STEP 2: Take the first item in the list and attempt to place it in a (randomly selected) box. If no items are in the list, the optimal solution has been found; stop. If sufficient space remains to accommodate the item in the box chosen, go to STEP 4: If insufficient space remains however go to STEP 3.

STEP 3: Attempt to place the item in one of the remaining boxes. If such a box exists, go to STEP 4; otherwise, go to STEP 5.

STEP 4: Record the assignment of the item and its weight to the box and remove the item from the list. Return to STEP 2.

STEP 5: List the set of boxes in descending order of remaining capacity. Let m_o be the box number for which minimum excess capacity remains. Considering boxes one and two in the list only, attempt a one-for-one exchange of items between boxes such that the available space in one of the boxes is fully utilized. If such an exchange can be made amongst all possible one-for-one exchanges, record it and return to STEP 2. If no such exchange is possible, attempt two-for-one, then one-for-two exchanges, box one to box two. If such an exchange is possible, record it and go to STEP 2; otherwise, go to STEP 6.

STEP 6: Repeat STEP 5 for boxes one and three, one and four, etc. to one and m_o, then two and three, two and four, etc. to two and m_o, and so on. If a satisfactory exchange is still not evident, terminate; the heuristic procedure does not yield a feasible solution to the original problem.

It is important to note that the above exchange routine will always produce a feasible solution to (P2) and if every item in S is successfully placed in boxes an optimal solution too. Of course when some items in S cannot be assigned to boxes, the solution may still be optimal, but unproven. Furthermore, for (P1) the exchange routine can be modified to always yield a feasible solution. The modification is that in STEP 6 when it seems not all items can be assigned to the boxes in T, T may be expanded to include a box not originally belonging to T. Again in such a case an optimal solution to the original problem (P1) may still be found but we cannot prove its optimality.

IV. COMPUTATIONAL EXPERIENCE AND CONCLUSIONS

The procedure as described has been programmed in Fortran V code and run on a UNIVAC 1110. While (P1) and (P2) represent different problem situations, the solution procedure we use for each is essentially the same, and the results obtained for one of the problems using our procedure would be expected to reflect closely the procedure's effectiveness in solving the other. For this reason, computational results for solving (P2) only will be presented. The single knapsack routine we use for solving the surrogate relaxation is adopted from Program β of Ahrens and Finke [1].

[1] The brackets [ ] denote the greatest integer less than or equal to the enclosed quantity.

A series of 100 problems consisting of up to 1000 objects and up to six boxes were obtained by generating values and weights independently from a uniform distribution in the interval [10, 100]. Box capacities were then generated in a similar manner except the interval $b_l \leqslant b_j \leqslant b_u$ was used where $b_l = [.4\ (\Sigma w_i/n)]$,[1] $b_u = [.6\ (\Sigma w_i/n)]$. The final box capacity generated, b_n, was chosen such that occupancy ratio $= \Sigma b_j/\Sigma w_i = .5$. If $b_j < \min_i w_i$ or $\max_j b_j < \max_i w_i$, the set of generated box capacities were discarded and a new set generated. The occupancy ratio of .5 was used for all problems attempted.

Table 1 indicates computation times for our algorithm when solving (P2). For each item/box combination a total of ten problems were attempted, and the total number of optimal solutions to (P2) using the exchange routine was recorded along with solution times. For all problems except one, the optimal solution was obtained upon the first application of the exchange routine. For the remaining problem, the exchange routine was rerun but objects were assigned to boxes according to a (new) random order, at which time an optimal solution was found. As an indication of the relative efficiency of the procedure, the algorithm of Hung and Fisk [10] using a Lagrangian relaxation solved, after 250 seconds CPU time (UNIVAC 1108), only four problems in a ten-problem set containing 200 items and four boxes. The procedure presented here was able to solve all ten of these problems in 4.6 seconds.

TABLE 1. *Computational Results for (P2)*

	Four Boxes		Six Boxes	
Number of Items	Solution Time*	Proven Optimal Solutions	Solution Time	Proven Optimal Solutions**
50	.03/.02/.05	10	.03/.03/.06	9
100	.09/.05/.14	10	.09/.05/.14	10
200	.35/.11/.46	10	.35/.13/.48	10
500	2.0/.6/2.6	10	2.0/.6/2.6	10
1000	7.9/2.1/10.1	10	7.9/2.1/10.1	10

*Average CPU seconds per problem for surrogate relaxation/exchange routine/total

**For the one problem not terminating optimally, a feasible solution was found whose solution value was within 1.5% of the surrogate bound

As can be seen from Table 1, the computational efficiency of the heuristic is less sensitive to the number of objects in the knapsack algorithm than are other algorithms [4], [10], [11], [12], and [13]. Furthermore, the exchange routine is much less sensitive to the number of boxes than are the other algorithms. As a further example of this, a set of ten problems generated as in Table 1 but having 1000 objects and 10 boxes was solved in about the same number of seconds total CPU time as required by the problem set having 1000 items and 6 boxes. All problems were again proven optimal. In effect, the overall efficiency of the heuristic is equivalent to that of a knapsack algorithm.

As a final test of our procedure, we chose to solve a series of problem sets containing 200 items and four boxes, generated as in Table 1 but having narrower ranges of item weights and

[1]The brackets [ ] denote the greatest integer less than or equal to the enclosed quantity.

box sizes. The results are summarized in Table 2. A total of ten problems were attempted for each set, and the number of problems for each range of item weights and box sizes which terminated optimally are specified.

Table 2 indicates that the exchange procedure remains effective as long as some variation exists amongst item weights. The amount of variation in box sizes seems to have little effect upon the ability of the procedure to terminate optimally. That all ten problems terminate optimally for the problem set in which no variability is allowed within either item weights or box sizes is apparently fortuitous. For those problems in Table 2 which did not terminate optimally, feasible solutions were obtained having solution values which were in every case within 2.1% of the surrogate bound.

TABLE 2. *Effect of Variation in Box and Item Weights Upon Ability to Terminate Optimally*

Range of Item Weights	Range of Box Sizes		
	$.4b_u \leqslant b_j \leqslant .6b_u$*	$.45b_u \leqslant b_j \leqslant .55b_u$	$b_j = .5b_u$
10-100	10**	10	10
25-85	10	10	10
40-70	10	10	10
55	0	0	10

*$b_u = \Sigma w_i/4$
**Number of problems terminating optimally

As seen from Tables 1 and 2, the exchange routine described here appears to be quite effective in obtaining provably optimal solutions to loading problems of the type (P1) or (P2) in which a number of items and boxes are large and at least some variation exists in item weights. For smaller problems, and for problems in which little or no variation in item weights exists, available optimizing procedures may be more appropriate.

REFERENCES

[1] Ahrens, J.H. and G. Finke, "Merging and Sorting Applied to the 0-1 Knapsack Problem," Operations Research, Vol. *23*, pp. 1099-1109 (1975).
[2] Balas, E., "An Additive Algorithm for Solving Linear Programs with Zero-One Variables," Operations Research, Vol. *13*, pp. 517-546 (1965).
[3] Balas, E., "Discrete Programming by the Filter Method," Operations Research, Vol. *19*, pp. 915-957 (1967).
[4] Eilon, S. and N. Christofides, "The Loading Problem," Management Science, Vol. *17*, pp. 259-268 (1971).
[5] Fisk, J., "An Initial Bounding Procedure for Use with 0-1 Single Knapsack Algorithms," Opsearch, Vol. *14*, pp. 88-98 (1977).
[6] Geoffrion, A., "An Improved Implicit Enumeration Approach for Integer Programming," Operations Research, Vol. *17*, pp. 437-454 (1969).
[7] Glover, F., "Surrogate Constraints," Operations Research, Vol. *16*, pp. 741-749 (1968).
[8] Glover, F., "Surrogate Constraint Duality in Mathematical Programming," Operations Research, Vol. *23*, pp. 434-451 (1975).
[9] Glover, F., "Heuristics for Integer Programming Using Surrogate Constraints," Decision Sciences, Vol. *8*, No. 1, pp. 156-166 (1977).

[10] Hung, M.S. and J.R. Brown, "An Algorithm for a Class of Loading Problems," Naval Research Logistics Quarterly, Vol. *25*, pp. 289-297 (1978).
[11] Hung, M.S. and J.C. Fisk, "An Algorithm for 0-1 Multiple Knapsack Problems," Naval Research Logistics Quarterly, Vol. *25*, pp. 571-579 (1978).
[12] Ingargiola, G. and J. Korsh, "An Algorithm for the Solution of 0-1 Loading Problems," Operations Research, Vol. *23*, pp. 1110-1119 (1975).
[13] Martello, S. and P. Toth, "Solution of Zero-One Multiple Knapsack Problems," presented at the ORSA/TIMS National Meeting, Atlanta (1977).

are applicable only when certain relationships obtain between the detection probabilities for the various regions.

In Section 3, a continuous-time version of the problem is described for which it transpires that $P^* = P_o$. This continuous-time version is asymptotically equivalent to the discrete-time version as the detection probabilities tend to zero.

Section 4 summarizes our investigation of the relationship between P^* and P_o for different values of N and of the detection probabilities. P_o is a particularly good approximation to P^* when the q_i $(i = 1, 2, \ldots, N)$ are either all sufficiently large, or all sufficiently small, or not too dissimilar. Furthermore, if the range of the q_i values is held more or less constant, the accuracy of the approximation does not vary greatly with N. The case of $N = 2$ may as a result be used as a point of reference, and a method of doing this is described.

Finally, in Section 5 we assess the N-region problem as viewed by the other player, the searcher. Although it transpires that the searcher's optimal strategy is likely to be difficult to determine exactly, this section shows that satisfactory approximations to it are usually determinable without too much difficulty.

2. THE DETERMINATION OF P^* AND $V(P^*)$

This section describes three distinct approaches to the evaluation of P^* and $V(P^*)$. The first is quite general and it may be used for any problem of N regions, where N is limited only by the capacity of one's computational facilities. There are no restrictions on the values of the escape probabilities. The accuracy of P^* and $V(P^*)$ which this approach yields can be made precise to any arbitrary degree.

The second method applies to problems which are multiples of smaller problems whose characteristics are already known. For example the problem

$$(r_1, r_2, r_3, r_4, r_5, r_6)$$

is related to the smaller problem (r_x, r_y) if r_1, r_2, r_3 are all equal to r_x, and r_4, r_5, r_6 are all equal to r_y: that is, the former problem consists simply of three blocks of the latter. So if P^* and $V(P^*)$ have been established for the smaller problem, this approach indicates how these same characteristics may be obtained for the larger problem.

The third approach is applicable to problems of any N where the escape probabilities assume only two values and where one escape probability is an integer power of the other. These conditions lead to particularly simple expressions for P^* and $V(P^*)$, and these expressions yield useful bounds for $V(P^*)$ in situations where such conditions are not satisfied.

(i) A general method

We begin by outlining a procedure for finding the expected payoff $V(P)$ at any vector P, by assuming that the searcher always plays optimally; that is, he consistently searches that region with the greatest current p_iq_i. To simplify the exposition, assume that if $i \neq j$, then $p_iq_i \neq p_jq_j$, both for the original vector P and for the vectors into which it is transformed by (1).

Suppose then, without loss of generality, that P is such that

$$p_1q_1 > p_2q_2 > \ldots > p_Nq_N.$$

Let b_{ijk} be the number of searches in region i before the k-th search of region j. From (2), assuming the searcher's policy is as specified above, we can write

$$p_i r_i^{(b_{ijk}-1)} q_i > p_j r_j^{k-1} q_j,$$

and

$$p_i r_i^{b_{ijk}} q_i < p_j r_j^{k-1} q_j.$$

Suppose x is such that

$$p_i r_i^{x} q_i = p_j r_j^{k-1} q_j.$$

Hence

$$x = \{\log(p_j q_j / p_i q_i) + (k-1) \log r_j\}/\log r_i$$

and

$$b_{ijk} - 1 < x < b_{ijk}.$$

We can therefore write

$$b_{ijk} - 1 = \text{Int}\left\{\frac{\log (p_j q_j / p_i q_i) + (k-1) \log r_j}{\log r_i}\right\}$$

(3)

$$= \text{Int}\left\{\frac{\log (p_j q_j / p_i q_i)}{\log r_i} + (k-1)\frac{n_i}{n_j}\right\},$$

where Int denotes the integer part, and we take r_i and r_j to be related by $r_i^{n_i} = r_j^{n_j}$, where n_i and n_j are any numbers, not necessarily integer.

Next we define

$$V_{ji} = E(\text{no of searches in region } j \mid \text{evader in region } i).$$

$$V_i = E(\text{total no of searches} \mid \text{evader in region } i)$$

$$= \sum_{j=1}^{N} V_{ji}.$$

From the definition of q_i, $V_{ii} = 1/q_i$. Otherwise, we can use the definition of b_{ijk} to write expressions for V_{ij} in the following two situations:

(a) $\underline{i < j}$

$$V_{ij} = b_{ij1} + r_j(b_{ij2} - b_{ij1}) + r_j^2 (b_{ij3} - b_{ij2}) + r_j^3 (b_{ij4} - b_{ij3}) + \ldots$$

$$= q_j(b_{ij1} + r_j b_{ij2} + r_j^2 b_{ij3} + \ldots).$$

To a first approximation, using (3),

$$b_{ijk} = b_{ij1} + (k-1)\frac{n_i}{n_j}.$$

So therefore

$$V_{ij} = b_{ij1} + q_j \frac{n_i}{n_j} \sum_{k=2}^{\infty} (k-1) r_j^{k-1}$$

$$= b_{ij1} + \frac{r_j}{q_j n_j} n_i.$$

THE SEARCH FOR AN INTELLIGENT EVADER CONCEALED IN ONE OF AN ARBITRARY NUMBER OF REGIONS

J.C. Gittins

University Mathematical Institute
Oxford, England

D.M. Roberts

Ministry of Defence
London, England

ABSTRACT

This paper considers the search for an evader concealed in one of an arbitrary number of regions, each of which is characterized by its detection probability. We shall be concerned here with the double-sided problem in which the evader chooses this probability secretly, although he may not subsequently move; his aim is to maximize the expected time to detection, while the searcher attempts to minimize it.

The situation where two regions are involved has been studied previously and reported on recently. This paper represents a continuation of this analysis. It is normally true that as the number of regions increases, optimal strategies for both searcher and evader are progressively more difficult to determine precisely. However it will be shown that, generally, satisfactory approximations to each are almost as easily derived as in the two region problem, and that the accuracy of such approximations is essentially independent of the number of regions. This means that so far as the evader is concerned, characteristics of the two-region problem may be used to assess the accuracy of such approximate strategies for problems of more than two regions.

. INTRODUCTION

In a recent paper — Roberts and Gittins [5], hereinafter referred to as R & G — an nalysis was given of a search problem involving two regions. The analysis is extended in this aper to problems of similar type but with an arbitrary number of regions. Such problems may e described as follows:

Suppose a stationary object is hidden in one of N distinct regions. The probability of its eing concealed in region i ($i = 1, 2, \ldots, N$) will be denoted by p_i, and the location probability vector by

$$P = (p_1, p_2, \ldots, p_N).$$

ach region is characterized by its detection probability q_i which is the probability that a search f region i will discover the object if it is there; to avoid unnecessary complications, suppose

$$0 < q_i < 1.$$

Often it will be advantageous to specify a region in terms of its escape probability r_i, where $r_i = 1 - q_i$. We assume that the time taken to search any region is constant, and take this constant to be the unit of time.

From Bayes' theorem it follows that an unsuccessful search of region j changes the location probability vector as shown below.

(1)
$$p_j \xrightarrow{j} \frac{p_j(1-q_j)}{1-p_jq_j},$$
$$p_i \xrightarrow{j} \frac{p_i}{1-p_jq_j} \text{ for all } i \neq j.$$

It has been shown by, among others, Black [1] that the strategy which at any time searches the region with the greatest current value of p_iq_i minimizes the expected time until the object is found. For a given initial P this minimum is denoted $V(P)$. Usually such a strategy is deterministically defined, and as such can be considered as pure. However there are occasions when p_iq_i is maximized for more than one value of i. Clearly in these circumstances the searcher can choose between pure strategies, each of which will lead to the minimum expected time to detection. We shall adhere to the terminology used in Norris [4] by referring to these pure strategies as 'good' strategies.

We can extend equation (1) to determine that the transformation due to a sequence of searches which involves a total of $\{k_i\}$ searches of region i, for each i, will be as follows:

(2)
$$p_i \xrightarrow{\{k_i\}} \frac{p_i r_i^{k_i}}{\sum_{j=1}^{N} p_j r_j^{k_j}}.$$

Significantly, the order of the sequence has no effect on the final transformation.

In R & G this single-sided problem was considered in the form of a double-sided search problem in which the initial value of P is chosen secretly by the object (or evader). In this form the problem is a zero-sum game between the searcher and the evader, the payoff to the evader being the time which the searcher takes to find him. This game has been considered by Bram [2] who showed that it does possess a value, and that therefore the appropriate strategies for the searcher and the evader are the minimax and maximin strategies respectively.

It is easy to show that if the evader is allowed to move quite freely between searches then his maximum strategy is given at each stage by the location probability vector P_o, defined such that p_iq_i is the same for all i. In R & G it was shown that when $N = 2$, P_o is a remarkably good approximation to the evader's maximum strategy P^* for the more complicated, and often more realistic, problem in which the evader, once hidden, remains stationary. This observation also led to nearly optimal search strategies. The significance of these approximations is that they are much more easily calculated than the exact solutions. In this paper we show that for arbitrary values of N the approximation $P^* = P_o$ remains extremely good under most circumstances, and this is our most important conclusion.

Detailed methods of calculating P^* to any desired accuracy are given in Section 2. One of these can be used for any set of detection probabilities. The others are abridged methods which

Similarly, to a second approximation,

$$b_{ijk} = b_{ij2} + (k-2)\frac{n_i}{n_j},$$

$$V_{ij} = q_j b_{ij1} + r_j\left(b_{ij2} + \frac{r_j}{q_j n_j}\, n_i\right).$$

So for the l-th approximation

$$V_{ij} = q_j b_{ij1} + q_j r_j b_{ij2} + q_j r_j^2\, b_{ij3} + \ldots + r_j^{l-1}\left(b_{ijl} + \frac{r_j}{q_j n_j}\, n_i\right)$$

(b) $\underline{i > j}$

$$V_{ij} = (r_j^{b_{ji1}} - r_j^{b_{ji2}}) \times 1 + (r_j^{b_{ji2}} - r_j^{b_{ji3}}) \times 2 + (r_j^{b_{ji3}} - r_j^{b_{ji4}}) \times 3 \ldots$$
$$= r_j^{b_{ji1}} + r_j^{b_{ji2}} + r_j^{b_{ji3}} + \ldots .$$

To a first approximation, again using (3),

$$V_{ij} = r_j^{b_{ji1}}\,(1 + r_i + r_i^2 + \ldots)$$
$$= \frac{r_j^{b_{ji1}}}{q_i}.$$

Similarly, to a second approximation,

$$V_{ij} = r_j^{b_{ji1}} + r_j^{b_{ji2}}\,(1 + r_i + r_i^2 + \ldots)$$
$$= r_j^{b_{ji1}} + \frac{r_j^{b_{ji2}}}{q_i}.$$

And for the l-th approximation

$$V_{ij} = r_j^{b_{ji1}} + r_j^{b_{ji2}} + \ldots . + \frac{r_j^{b_{jil}}}{q_i}.$$

Thus we have established a series of successive approximations for $V_i(P)$, depending on the value of l up to which b_{ijl} is given its exact integer value. Since

$$V(P) = \sum_i p_i V_i(P) \tag{4}$$

we can thus calculate $V(P)$ to any desired accuracy.

In an examination of problems of various sizes, it was found that $l = 10$ gave very good results indeed — typically resulting in a $V(P)$ accurate to within $\pm\ 10^{-4}$ of its true value. If $l = 20$ is used, then understandably $V(P)$ is much closer to its true value; it was always observed to be within $\pm\ 10^{-5}$, and frequently far closer.

Having found a precise method of calculating $V(P)$ it is necessary next to find an approach to evaluating P^* and hence $V(P^*)$. Since P_o is always easy to calculate, and is always close to P^* if not actually equal to it, an obvious approach is to proceed as follows. Starting

from the vector P_o, and having calculated $V(P_o)$, form a new vector P by increasing its first component by, for example, 0.01, and decreasing each of its remaining components by an amount proportional to the component's magnitude. If $V(P) > V(P_o)$ form a new vector by modifying P as P_o was modified. Continue thus until a maximum is reached. If $V(P) < V(P_o)$, search in the opposite direction. When a maximum is reached, at P' say, (or if no payoff greater than $V(P_o)$ is found in either direction) move from there (P' or P_o) by changing the vector's second component, and so on. When all directions have been examined, the entire procedure can be repeated using progressively smaller increments until an acceptable P^* is found.

There are two reasons why this procedure invariably leads to an acceptable P^*. First, using an obvious extension to the argument in Section 2 of R & G, it may be shown that $V(P)$ is continuous and a concave function of P; hence, the problem of a local maximum does not arise. Second, $r_i < 1$ for all i, and this means that P^* is an interior maximum point; that is, each of its components is positive.

In the illustrative examples of Sections 4 and 5 the procedure was continued to the extent of using increments of 0.0001. When this sensitivity was combined with a value of $l = 30$ in determining $V(P)$ one arrived at a value of $V(P^*)$ which, whenever checked against known $V(P^*)$ (determinable as described in Sections 2(ii) or 2(iii)), was accurate to within 0.00004.

This procedure can prove to be inefficient if several regions have the same escape probability. In such cases it seems advisable to increment all the components of P associated with these particular regions together. This keeps such components equal, not an unreasonable policy when one considers that at P^* they will of course be equal. It may be advisable to observe similar precautions also when several regions have approximately equal escape probabilities.

When the incidence of regions with the same escape probability is such that the search problem may be regarded as the result of the joining together of a set of sub-problems, this modification to our general procedure may be linked with a simplified method of calculating $V(P)$. This situation will be discussed in Section 2(ii).

(ii) Problems with identical blocks of regions

Consider a problem for which there is an integral number k of blocks of regions, where the blocks (each of M regions, say) are identical in all respects. The location probability vector P^+ for such a problem may be expressed, with an obvious notation, in the form

$$P^+ = (p_{ij}^+ : i = 1, 2, \ldots, M; j = 1, 2, \ldots, k).$$

The expected payoff if the evader uses the strategy P^+ and the searcher plays so as to minimize the expected time to detection will be denoted by $V_{kM}(P^+)$. We shall use $V_M(P)$ to designate the similar function for the reduced problem in which the evader is restricted to one particular block of M regions and plays the vector $P = (p_i; i = 1, 2, \ldots, M)$. We shall say that P^+ *corresponds* to P if

$$p_{ij}^+ = p_i/k, \quad i = 1, 2, \ldots, M; \quad j = 1, 2, \ldots, k.$$

In such cases the calculation of $V_{kM}(P^+)$ is simplified by the following result.

THEOREM 1: If P^+ corresponds to P then

$$V_{kM}(P^+) = k\left\{V_M(P) - \frac{1}{2}\left[1 - \frac{1}{k}\right]\right\}.$$

PROOF: The expected number of searches when the evader is hidden in the j th block is $kV_M(P) - (k - j)$. The probability that the evader is actually hidden in the j th block is clearly $1/k$. Hence

$$V_{kM}(P^+) = \sum_j \frac{1}{k} \{kV_m(P) - (k - j)\}$$

$$= k \left\{ V_M(P) - \frac{1}{2} \left[1 - \frac{1}{k} \right] \right\}.$$

COROLLARY: P^{*+} corresponds to P^* and

$$V_{kM}(P^{*+}) = k \left\{ V_M(P^*) - \frac{1}{2} \left[1 - \frac{1}{k} \right] \right\}.$$

Here P^{*+} and P^* are the evader's maximum strategies for a k-block problem and the related single-block problem respectively. The proof is immediate from Theorem 1 and from the observation that p^{*+}_{ij} must be independent of j for all i. It follows that the values of P^* and $V(P^*)$ for any M-region problem may be used to calculate similar quantities for problems defined by adding identical blocks of M regions.

(iii) Problems with just two escape probabilities, one a power of the other

Consider a problem with regions (1, 1), (1, 2), . . , (1, A), (2, 1), (2, 2), . . , (2, B), with

$$q_{11} = q_{12} = \ldots = q_{1A} = q_1 = 1 - r_1,$$

$$q_{21} = q_{22} = \ldots = q_{2B} = q_2 = 1 = r_2,$$

and with $r_1^n = r_2$, where n is an integer.

The components of P_o are

$$p^o_{1i} = q_2/(Aq_2 + Bq_1), \; i = 1, 2, \ldots, A,$$

$$p^o_{2i} = q_1/(Aq_2 + Bq_1), \; i = 1, 2, \ldots, B.$$

For a problem of this type, the infinite series which, as described in Section 2(i), arise in the calculation of $V(P_o)$ may be summed explicitly, leading to the expression

$$V(P_o) = \frac{\frac{1}{2}A(A+1)q_2 + \frac{1}{2}B(B+1)q_1 + ABq_2 + A^2r_1 \left(\frac{q_2}{q_1} - nr_1^{n-1} \right)}{Aq_2 + Bq_1} + (An + B)\frac{r_2}{q_2}.$$

Clearly the point P* must be such that

$$p^*_{1i} = p^*_{1j}, \; 1 \leqslant i,j \leqslant A,$$

and

$$p^*_{2i} = p^*_{2j}, \; 1 \leqslant i,j \leqslant B.$$

Closed expressions may also be obtained for $V(P)$ for other vectors P with these properties. Consider for example the vector P' with components

$$p'_{1i} = q_2/(Aq_2 + Bq_1r_1), \; i = 1, 2, \ldots, A,$$

$$p'_{2i} = q_1r_1/(Aq_2 + Bq_1r_1), \; i = 1, 2, \ldots, B.$$

Thus the strategy for the searcher which minimizes the expected time to detection corresponding to the evader's having chosen to play P' starts with searches of each of the first A regions, and it follows from equation (2) that this transforms the vector P' into the vector P_o. We have

$$\begin{aligned} V(P') = {} & P(\text{evader is found in one of the first } A \text{ searches}) \times \\ & E(\text{no. of searches required} \mid \text{evader is found in one of the first } A \text{ searches}) + \\ & P(\text{evader is not found in one of the first } A \text{ searches}) \times \\ & E(\text{no. of searches required} \mid \text{evader is not found in one of the first } A \text{ searches}) \\ = {} & Ap'_{11}q_1 \times \frac{1}{2}(A+1) + (Ap'_{11}r_1 + Bp'_{21}) \times (V(P_o) + A). \end{aligned}$$

Similar closed expressions may be obtained for $V(P)$ for all those vectors P which have the symmetries stated above, and which may be transformed into the vector P_o. These are the values of P which are of most interest, since P^* must be one of them. This may be shown along the lines described in R & G for the case when $N = 2$.

3. THE SEARCH PROBLEM IN CONTINUOUS TIME

The N regions are now characterized by their detection rates λ_i $(i = 1, 2, \ldots, N)$. As before the evader may not move, but the location probability vector changes by Bayes' theorem as the search progresses. Its value at time t is denoted by $(p_1(t), p_2(t), \ldots, p_N(t))$ and P_o is the vector for which $p_i\lambda_i$ is a constant.

In continuous time it is natural to allow the searcher to divide his effort at any given time between the N regions. If $u_i(t)$, a piecewise continuous† function of t, is the proportion of his total effort allocated to region i at time t $[u_1(t) + u_2(t) + \ldots + u_N(t) = 1]$, the probability of the evader being detected in the time interval $(t, t + \delta t)$ if he is in region i, and conditional on detection not occurring before time t, is

$$\lambda_i u_i(t)\,\delta t + o(\delta t), \tag{5}$$

except at a point of discontinuity of the function $u_i(t)$. The probability of detection in $(t, t + \delta t)$ if the location of the evader is given by the initial probability vector, and conditional on detection not having occurred before time t, is therefore

$$\sum_{i=1}^{N} \lambda_i u_i(t) p_i(t)\,\delta t + o(\delta t); \tag{6}$$

and for the sake of convenience, we shall write this in the form $\rho(t)\delta t + o(\delta t)$. The expressions (5) and (6) lead respectively, and by elementary arguments, to alternative expressions for $\mathcal{F}(t)$, the probability that detection does not take place before time t. We have

$$\mathcal{F}(t) = \sum_{i=1}^{N} p_i(0)\,\exp[-\lambda_i U_i(t)] = \exp\left[-\int_0^t \rho(s)\,ds\right], \tag{7}$$

where

$$U_i(t) = \int_0^t u_i(s)\,ds.$$

Also if T is the time to detection it is well known, and easily shown by integration by parts, that

$$E(T) = \int_0^\infty \mathcal{F}(t)\,dt. \tag{8}$$

†It is not difficult to modify the argument so that $u_i(t)$ is required only to be a measurable function of t.

As for the discrete-time case, the evader's strategy is

$$P = (p_1(0),\ p_2(0),\ \ldots,\ p_N(0)).$$

The searcher's strategy is the vector function u which for any $t > 0$ takes the value $(u_1(t),\ u_2(t),\ \ldots,\ u_N(t))$. We shall denote the evader's maximum strategy by P^* and proceed now to prove -

THEOREM 2: For the continuous time problem $P^* = P_o$.

PROOF: We note firstly that it follows from equations (7) and (8) that

$$E(T) = \int_0^\infty \sum_{i=1}^{N} p_i(0)\ \exp[-\lambda_i\ U_i(t)]\ dt.$$

It may be shown that this expression is minimized if and only if the searcher uses the continuous time version of the rule that he should search the region with the greatest current $p_i q_i$; specifically —

$$u_i(t) > 0 \iff \lambda_i\ p_i(t) = \max_i \lambda_j p_j(t)$$

except possibly at a set of times of Lebesgue measure zero. This result follows from a variational argument similar to that used by Gittins [3] for another resource allocation problem. Alternatively it may be established using the concept of uniform optimality discussed by Stone [6].

Under such a rule there is clearly some $t_1 (< \infty)$ such that

$$t_1 = \inf\ \{t:\ \lambda_1 p_1(t) = \lambda_2 p_2(t) = \ldots = \lambda_N p_N(t)\},$$

and

$$p_i(t) = \lambda_i^{-1} \Big/ \sum_{j=1}^{N} \lambda_j^{-1}, \quad i = 1,\ 2,\ \ldots,\ N;\ t \geqslant t_1. \tag{9}$$

From Bayes' theorem we have

$$p_i(t) = p_i(0)\ \exp[-\lambda_i\ U_i(t)] \Big/ \sum_{j=1}^{N} p_j(0)\ \exp[-\lambda_j\ U_j(t)],$$

$$i = 1,\ 2,\ \ldots,\ N;\ t \geqslant 0,$$

so that

$$\lambda_1 p_1(0)\ \exp[-\lambda_1\ U_1(t)] = \lambda_2\ p_2(0)\ \exp[-\lambda_2\ U_2(t)] = \ldots$$

$$= \lambda_N\ p_N(0)\ \exp[-\lambda_N\ U_N(t)], \quad t \geqslant t_1. \tag{10}$$

Now differentiating equation (10) with respect to t, and dividing by (10), gives

$$u_i(t) = \lambda_i^{-1} \Big/ \sum_{j=1}^{N} \lambda_j^{-1}, \quad i = 1,\ 2,\ \ldots,\ N;\ t \geqslant t_1.$$

Thus from (6),

$$\rho(t) = \left(\sum_{j=1}^{N} \lambda_j^{-1}\right)^{-1}, \quad t \geqslant t_1. \tag{11}$$

A similar argument shows that

(12) $$\rho(t) > \left(\sum_{j=1}^{N} \lambda_j^{-1}\right)^{-1}, \quad t < t_1.$$

From equations (7), (8), (11), (12) it follows that if the searcher uses a continuous time version of the principle that in order to minimize the expected searching time, he should search that region with the greatest current $p_i q_i$, then

$$E(T) \leqslant \sum_{j=1}^{N} \lambda_j^{-1},$$

with equality if and only if $t_1 = 0$. Now $t_1 = 0$ if and only if $p_i(0)\lambda_i$ is a constant; that is to say if and only if $P = P_o$. Thus the evader's maximum strategy is P_o and the theorem is proved.

4. STRATEGIES FOR THE EVADER

The Two-Region Problem and Related, Larger Problems

From an examination of even a few N-region problems, one very soon concludes that several of their characteristics depend on the value of N. Any description is therefore perhaps best undertaken in terms of increasing N. And we shall begin with the simple two region problem, $r_1 = 0.8$, $r_2 = 0.512$. (This problem has, as it happens, been chosen such that

$$r_1^3 = r_2$$

but, as we shall see later, this has not in any way affected the generality of the conclusions).

For this particular problem there are no fewer than three ways of finding $V(P_o)$, $V(P^*)$ and P^*. (P_o of course is always simply determined from $p_i q_i$ being a constant). First we note that the method of Section 2(iii) may conveniently be followed. Second, the general approach of Section 2(i) can be used, although usually we cannot expect to arrive at an exact $V(P_o)$ (though in this particular case, we do, fortuitously) nor an exact P^* and $V(P^*)$. Third, as in very many two-region problems (see R & G) a dynamic programming method, that is first described in Norris [4], is entirely feasible.

So for this problem, we have

$$V(P_o) = 6.51067\,,$$
$$V(P^*) = 6.53006\,,$$
$$\frac{V(P^*)}{V(P_o)} = 1.00298\,.$$

If in our examination of the effects of increasing N we see this as a starting point, then clearly we can look at a whole range of four-region problems, all *related* to this basic two-region problem in the sense that the largest and the smallest of the four escape probabilities are 0.8 and 0.512. It is often convenient to classify N-region problems in terms of the largest and smallest escape probabilities; we shall refer to these as r_l and r_s respectively.

Four of these problems are listed in Table 1.

TABLE 1.

EXAMPLE	r_1	r_2	r_3	r_4	$V(P_o)$	$V(P^*)$	$V(P^*)/V(P_o)$
1.1	0.8	0.8	0.8	0.512	15.501	15.526	1.00160
1.2	0.8	0.8	0.512	0.512	12.521	12.560	1.00310
1.3	0.8	0.512	0.512	0.512	9.574	9.609	1.00362
1.4	0.8	0.7	0.6	0.512	11.312	11.327	1.00127

Three features are immediately evident from the table:

(i) The largest $V(P^*)/V(P_o)$ exceeds the corresponding ratio in the two-region problem.

(ii) This occurs in a problem where only the largest and the smallest escape probabilities are present — example 1.3 (problems where r_2, r_3 are distributed between r_1 and r_4 have $V(P^*)/V(P_o)$ ratios which are very much less — see for example 1.4).

(iii) Example 1.2 consists of two blocks of our original two-region problem. For this example $V(P^*)$ and $V(P_o)$ — as well as of course P^* — could have been derived from the corresponding characteristics of the two-region problem using Theorem 1.

These features continue to manifest themselves as N increases; to eight for instance, as in Table 2. (Unlike Table 1, this table does not contain all the frequencies of r_l and r_s; however the example for which $V(P^*)/V(P_o)$ takes its maximum value has been included).

TABLE 2.

EXAMPLE	r_1 r_5	r_2 r_6	r_3 r_7	r_4 r_8	(VP_o)	(VP^*)	$\frac{V(P^*)}{V(P_o)}$
2.1	0.8 0.8	0.8 0.8	0.8 0.8	0.8 0.512	33.498	33.525	1.00081
2.2	0.8 0.8	0.8 0.8	0.8 0.512	0.8 0.512	30.503	30.553	1.00163
2.3	0.8 0.512	0.8 0.512	0.8 0.512	0.8 0.512	24.543	24.620	1.00316
2.4	0.8 0.512	0.8 0.512	0.512 0.512	0.512 0.512	18.649	18.718	1.00372
2.5	0.8 0.65	0.75 0.6	0.7 0.55	0.65 0.512	21.188	21.198	1.00047

Specifically, the largest $V(P^*)/V(P_o)$ ratio continues to exceed the corresponding ratio in the two-region problem; also it has increased slightly. Moreover, it occurs as previously in a problem where the only escape probabilities present are either r_l or r_s — example 2.4 of Table 2. Once again we find some eight-region problems corresponding to four-region problems, or even, in one case, to our basic two-region problem (examples 2.4, 2.3, 2.2 to 1.3, 1.2, 1.1 respectively; example 2.3 is also composed of four blocks of the two-region problem).

If we look once again at example 2.4 we will see that it consists of two identical blocks each of four regions, where each block is that of example 1.3. If now we go on to consider the 12-region problem

$$r_1 = r_2 = r_3 = 0.8,$$

(A)

$$r_4 = r_5 = \ldots = r_{12} = 0.512,$$

then using Theorem 1 (noting that this problem consists of three identical blocks) we can determine that

$$\frac{V(P^*)}{V(P_o)} = 1.00375.$$

Significantly as the number of blocks increases still further this ratio will certainly increase, but will never exceed 1.00382. Likewise from our basic two-region problem we can infer that for all problems of N regions, where half of the regions have escaped probabilities 0.8, and the other half 0.512, $V(P^*)/V(P_o)$ is bounded above by 1.00322.

From Tables 1 and 2 we note that those examples with the largest ratios $V(P^*)/V(P_o)$ both have three times as many regions with escape probability 0.512 as with 0.8 (examples 1.3, 2.4). This seems to be fortuitous. For in the 12-region problem the largest $V(P^*)/V(P_o)$ ratio occurs in the example

$$r_1 = r_2 = r_3 = r_4 = 0.8,$$

(B)

$$r_5 = \ldots = r_{12} = 0.512,$$

although its value (1.00378) exceeds only slightly that of example (A).

The N-Region Problem — General Conclusions

Were we to have started with any other pair of escape probabilities as our basic two-region example, and proceeded to examine associated problems up to 12 regions, then almost certainly we could have observed features similar to (i), (ii) and (iii) above. In the course of this study such a procedure was carried out about a dozen times on a set of problems intended to give as representative a picture as possible. In each case this observation was found to apply, and, as might be expected, such an analysis enables one to be more explicit in one's conclusions.

Specifically, the largest $[V(P^*) - V(P_o)]/V(P_o)$ increases with increasing N. It exceeds that of the two-region problem, but is generally less than twice as large. Exceptions do exist, although they seem to be confined to problems where one (and sometimes, though rarely, two) escape probabilities are large (>0.85 say) and all the rest are by comparison quite small (about 0.1 or less); for example in the 12-region problem

$$r_1 = 0.95,\ r_2 = \ldots = r_{12} = 0.1285,$$

$V(P^*)$ is about 4½% greater than $V(P_o)$, whereas in the related two-region problem

$$r_1 = 0.95,\ r_2 = 0.1285,$$

$V(P^*)$ is about 1% greater than $V(P_o)$.

In two-region problems where $V(P^*)$ and $V(P_o)$ are equal, one finds that in larger, related problems, the same generally applies. Although this was not found to be invariably so, such differences as were observed were always of the order of 0.2% or less.

We are therefore able to say something about the ratio of $V(P^*)$ to $V(P_o)$ in N-region problems (where N takes values up to 12 at least) provided we possess certain information about the related two-region problem. The sort of information required is shown in Figure 1. We can see there that the two-region problem

$$r_1 = 0.8,\ r_2 = 0.2$$

possesses a $V(P^*)/V(P_o)$ ratio of about 1.015. What can we say then about the 12-region problem

$$r_1 = 0.8 > r_2 > \ldots > r_{11} > r_{12} = 0.2 \quad ?$$

First, if $r_2, \ldots, r_{11}$ are distributed at all within this interval, then the ratio of $V(P^*)$ to $V(P_o)$ is almost certain to be very much closer to one. Second, all such 12-region problems — even those consisting solely of the two extreme escape probabilities r_l and r_s — will have a $V(P^*)$ within 3% of $V(P_o)$; as indicated above such exceptions to this conclusion as do exist tend to be well defined, and this is certainly not one of them.

In any class of problems of the same size N, and characterized by the same r_l and r_s, the problem with the largest proportional difference between $V(P^*)$ and $V(P_o)$ will be one of those $N-1$ problems whose escape probabilities are r_l and r_s only; for instance example 1.3 of Table 1. What justification is there for proposing this conjecture? First, considerations of symmetry assure us that for any N-region problem with *all* regions having the same escape probability, P_o and P^* are the same. (One could call this a perfectly balanced problem). Second, where several escape probabilities are involved and where r_l and r_s are respectively the largest and the smallest of them, it can be seen that a "less balanced" problem can be created (i.e., one moves further away than one already was from the "all escape probabilities equal" situation) by moving some or all of those not equal to r_l or r_s to one extreme, with the remainder being moved to the other extreme. This conjecture has been examined in a number of cases, and has been found always to be correct.

Special Relationships Existing Between Escape Probabilities

All of the conclusions discussed so far were described using as a basis examples where the largest and smallest escape probabilities (r_l and r_s) were such that the ratio of the logarithms of r_l and r_s is an integer. In Table 1 for instance

$$r_l^3 = r_s.$$

Inevitably this inclines one to ask whether such a relationship has affected our conclusions in any way at all.

This question has been considered quite extensively, but no evidence has been found to suggest that such relationships have any influence. The behaviour of $V(P^*)/V(P_o)$ as a function of the escape probabilities has been explored where no integer relationship exists between these probabilities. The function's general characteristics were found to be indistinguishable from those observed where integer relationships do exist.

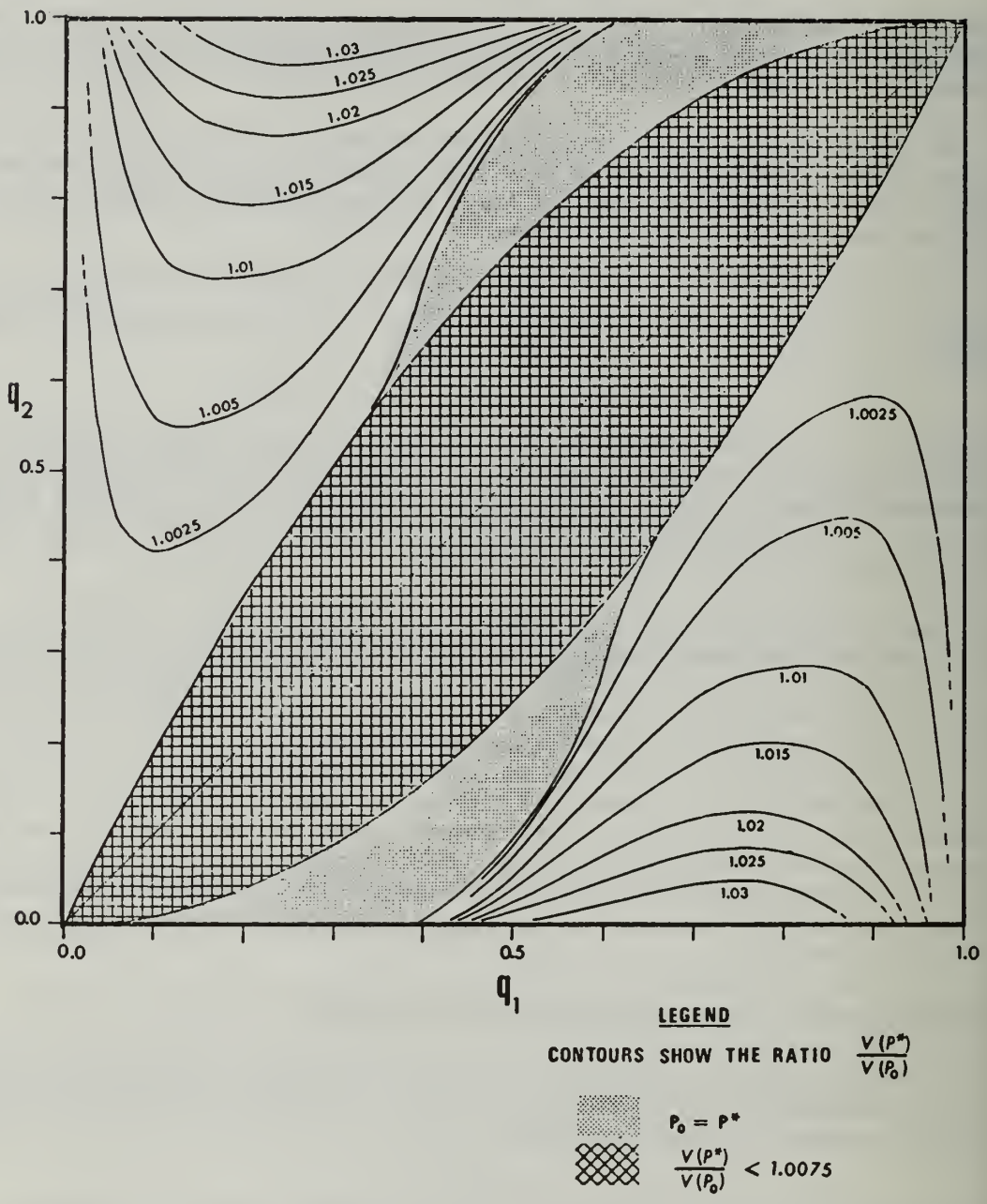


FIGURE 1. The two-region problem

5. STRATEGIES FOR THE SEARCHER

We have confined our attention so far to an analysis of what the evader should do in particular situations. We saw that very often he can be satisfied with merely calculating P_o, knowing that $V(P_o)$ is so close to $V(P^*)$ as to be adequate for his purposes. This suggests that the searcher might apply one of the good strategies vis à vis P_o. While this would often be a satisfactory procedure, such a strategy generally turns out to be less adequate than is P_o for the evader. However the searcher may always proceed via the calculations described in Section 2(i).

In making these calculations he will of course determine an arbitrarily precise approximation to P^* (which for convenience we shall refer to as P^+). However he will also have determined the components $V_i(P^+)$ from which, using equation (4), $V(P^+)$ is derived: viz

$$V(P^+) = \sum p_i^+ V_i(P^+).$$

A typical set of components for an eight-region problem is shown in Table 3.

TABLE 3.

i	1	2	3	4	5	6	7	8
r_i	0.8	0.75	0.7	0.65	0.65	0.6	0.55	0.512
$V_i(P^+)$	21.2907	21.031	21.1195	21.7895	20.7895	21.1753	21.1265	21.2774

$V_i(P^+)$ is defined in Section 2(i) as the expected payoff assuming that the evader is actually hiding in region i; it is shown as a function of P^+ merely to signify that it corresponds to a good strategy on the searcher's part vis à vis an evader strategy P^+. This search strategy happens to be pure, for $V_i(P^+)$ was calculated on that basis. Significantly, at P^* itself there will be several of these good strategies. The searcher's minimax strategy is one which ensures that the expected payoff is not greater than the value of the game, irrespective of what strategy the evader has played. It is a mixed strategy obtained by randomizing over those strategies which are good against P^*. The randomization yields a set of $V_i(P^*)$ each equal to the value of the game $V(P^*)$.

For the example of Table 3, $V(P^+)$ is 21.1985. It is interesting to note by how little any of the $V_i(P^+)$ differ from $V(P^+)$. In particular the largest of them (21.7895) is only 3% greater than $V(P^+)$, which by the definition of P^* is less than or equal to $V(P^*)$. This means that if the searcher plays the good strategy vis à vis P^+, then he limits the evader to a maximum expected pay-off of 21.7895, whatever strategy the latter happens to have chosen.

The situation we have just described where the largest $V_i(P^+)$ is only a few percent greater than $V(P^+)$ is quite common, and seems invariably to be the rule where the escape probabilities are well spread out between r_l and r_s as in Table 3. (This is in contrast to the somewhat clustered form of examples 2.3 and 2.4 of Table 2).

Should this situation not apply, then the searcher might be advised to consider points in the vicinity of P^+ with a view to finding one where it does. Significantly also, having calculated the $V_i(P)$ for several values of P, there is always the possibility that by randomizing over some of the associated pure strategies, he can effectively decrease still further the variation with respect to i. The reason for this is that if the searcher plays the good strategy associated with the vector P_j with probability $\pi_j (j = 1, 2, \ldots, M)$, then the expected time to detection, given that the evader is in region i, is

$$\sum_{j=1}^{M} \pi_j V_i(P_j).$$

Randomizing over different good strategies associated with the same vector P has a similar effect.

We can see this by referring once again to Table 3. The pure strategy corresponding to P^+ searches region 4 prior to region 5 even though $r_4 = r_5$; this accounts for the difference of

one between $V_4(P^+)$ and $V_5(P^+)$. But we could equally well have chosen to search region 5 before region 4. By selecting either pure strategy with probability 0.5 the searcher can effectively modify Table 3 so that

$$V_4(P^+) = V_5(P^+) = 21.2895,$$

thereby limiting the evader to a maximum possible payoff (of 21.2907 if the latter is actually hiding in region 1) within 0.4% of $V(P^+)$, and of course even closer to $V(P^*)$.

The procedure outlined in Section 2(i) can be divided into two parts. The first described how one could find the payoff $V(P)$ corresponding to any evader strategy P; the second suggested how by beginning with P_o and iterating one could converge towards P^*. In discussing possible policies for the searcher, we have so far been assuming that he actually follows this procedure to the extent of determining a P^+ near to P^*, and then examines the values of $V_i(P)$ at P^+ and in its vicinity. However, a large number of iterations is often needed to reach P^+. This immediately suggests a simplified approach open to the searcher. It is as follows:

Start with P_o and, using the first part of the procedure of Section 2(i), find $V(P_o)$. Next take a sample of vectors (N say) evenly distributed about P_o, and for each of these find $V(P)$. From this sample select a P for which the components $V_i(P)$ vary as little as possible. We have

$$V(P) \leqslant V(P^*) < \max_i V_i(P).$$

Clearly if $\max V_i(P) - V(P)$ is small, the searcher may feel justified in playing a good strategy vis à vis P.

Otherwise, of course, he can proceed to the evaluation of P^+, and then derive his strategy as described above.

REFERENCES

[1] Black, W.L., "Discrete Sequential Search," Information and Control, Vol *8*, pp 159-162 (1965).

[2] Bram, J., "A 2-Player *N*-Region Search Game," IRM-31, Operations Evaluation Group, Center for Naval Analysis, Washington (Jan 1963).

[3] Gittins, J.C., "Optimal Resource Allocation in Chemical Research," Advances in Applied Probability, Vol *1*, pp 238-270 (1969).

[4] Norris, R.C., "Studies in Search For a Conscious Evader," MIT Lincoln Laboratory Technical Report Number 279 (1962).

[5] Roberts, D.M. and J.C. Gittins, "The Search for an Intelligent Evader: Strategies for Searcher and Evader in the two-region problem," Naval Research Logistics Quarterly, Vol *25*, No. 1 (Mar 1978).

[6] Stone, L.D., *Theory of Optimal Search* (Academic Press, New York and London; 1975).

THE QUEUEING SYSTEM $M^X/G/1$ AND ITS RAMIFICATIONS

M. L. Chaudhry

Royal Military College of Canada
Kingston, Ontario, Canada

ABSTRACT

This paper deals with the bulk arrival queueing system $M^X/G/1$ and its ramifications. In the system $M^X/G/1$, customers arrive in groups of size X (a random variable) by a Poisson process, the service times distribution is general, and there is a single server. Although some results for this queueing system have appeared in various books, no unified account of these, as is being presented here, appears to have been reported so far. The chief objectives of the paper are (i) to unify by an elegant procedure the relationships between the p.g.f.'s

$$P(z) = \sum_{n=0}^{\infty} P_n z^n \text{ and } P^+(z) = \sum_{n=0}^{\infty} P_n^+ z^n$$

where P_n and P_n^+ are the limiting probabilities of queue lengths being n at random and departure epochs respectively, (ii) to correct an error in the paper by Krakowski and generalize his results and (iii) to discuss some other interesting cases of the system $M^X/G/1$ and its special cases.

INTRODUCTION

Several authors have discussed one aspect or the other of the queueing $M^X/G/1$ in which groups of random size, X, following a Poisson process join the system and are served individually by a single server whose service time distribution is general, e.g., Gaver [6], using renewal theoretic arguments, discusses, among other things, $P(z)$, the probability generating function (p.g.f.) of the limiting distribution of the number in the system at a random point in time; Sahbazov [14], using the imbedded Markov chain technique discusses $P^+(z)$, the p.g.f. of the number in the system at a departure epoch and the waiting time distribution for a random customer of an arrival group. The waiting time distribution discussed by Sahbazov [14] and some other authors seems to have been incorrectly reported; the correct formulation of which has recently been given by Burke [1]. It may be pointed out here that it is possible to derive $P^+(z)$ either from the paper of Harris [9], some results of which are reported in section 5.1.10 of Gross and Harris [7] or from the works of some other authors.

In queueing problems, among other things, the main interest, in general, centers around getting P_n. However to simplify the analysis, Kendall, in his two important papers, proposed the use of imbedded Markov chains and the researchers then got P_n^+ or other such probabilities by appropriately defining the regeneration points. Later, the other researchers in order to get P_n first got P_n^+ and then related P_n^+ to P_n. However, one can easily get P_n directly in many single server queueing systems with bulk arrival or bulk service or both, and if need be P_n^+ from

P_n. It is towards this purpose that this paper is chiefly addressed. To do this, we consider the queueing system $M^X/G/1$. The procedure has been successfully employed by Chaudhry and Templeton [2] in getting similar results for the bulk service queueing system $M/G^B/1$ wherein the server has a maximum capacity B.

Gross and Harris [7] relate the probabilities P_n to different types of imbedded Markov chain probabilities π_n using the semi-Markov approach. Their probabilities π_n are different from the probabilities P_n^+ (defined more accurately later) in that π_n's are considered at departure and arrival epochs when the system is empty, whereas P_n^+'s are considered only at departure epochs. Thus, clearly the set of epochs of the imbedded chain considered in this paper is a subset of the set of epochs of the imbedded chain considered by Gross and Harris [7]. Although the probabilities P_n are given in Gross and Harris, the technique of connecting P_n's and P_n^+'s is new and elegant in that one does not have to first obtain π_n by the imbedded chain as defined in Gross and Harris and then use the theory of semi-Markov processes or renewal theoretic arguments to get P_n—a normal practice so far. The method discussed here is new in that we reverse the procedure of first getting P_n^+ (or π_n) and then P_n, i.e., we first get $P(z)$ and then the relation between the p.g.f.'s of P_n and P_n^+ follows immediately and consequently the relations between various moments of the underlying distributions. Such relations between the moments neither appear to have been explicitly discussed in the literature nor would they be easily obtainable even if one tried to use the relation given in Gross and Harris [7] the exception to this statement being a recent result reported in Krakowski [11], the details of which are given in the next paragraph. It should, perhaps, be emphasized here that though the technique of supplementary variable is standard, its full impact is not yet known as will be revealed through the results that are being presented in this paper. Besides, the technique is more powerful than the other techniques discussed so far, because by using the procedure discussed in this paper one can, as mentioned earlier, even get similar results for several other single server queueing systems with bulk service or possibly bulk arrival and bulk service in which service time distribution is general. Obtaining results, through other techniques, similar to the ones that are being presented here for the latter type of queueing systems would be much more difficult, if not impossible, than through the procedure discussed here.

In a recent paper Krakowski [11] finds the average queue size at three epochs of time—random, just before arrival, and just after departure, for several queueing systems. However, for the system $M^X/G/1$, he obtains using intuitive arguments, the above averages only at an arrival epoch or at a random epoch (see scholion to his theorem B or section 4). Unfortunately, his result (S.23) has been incorrectly reported. Krakowski's [11] equation (S.23) is correct, if he was considering groups of constant sizes, but it does not appear to be so. By using mathematically more sound arguments, we later give correct expression for Krakowski's result (S.23). In fact, by our method, one can not only find the relations between averages, but also relations between higher order moments for the underlying distributions.

It may be appropriate here to mention the other related works which have been discussed in the literature. Foster and Perera [5], using renewal theoretic arguments, have discussed relations between steady-state probability generating functions (p.g.f.'s) of queue size considered at the above three epochs of time for the system $GI^r/M/1$ wherein customers following a recurrent process arrive in batches of fixed size r, and are served by a single server whose service time density is $b(x)$ such that

$$b(x) = \mu e^{-\mu x}, \quad x > 0, \quad \mu > 0.$$

Foster [4] also establishes heuristically some relations for the more general system $GI^r/G/1$. However, no such relations have systematically been reported for the system $M^X/G/1$ in which

the size of arrival groups is random—an assumption that would be better in many situations then the one when the size of arrival groups is constant.

In this paper, we carry out an analysis similar to the one carried out by Chaudhry and Templeton [2], for the queueing system $M^X/G/1$ wherein groups of random size, X, following a Poisson process join the system and are served individually by a single server whose service time distribution is general. In this way, we can, in principle, not only obtain relations between all the moments of queue size (for the queueing system $M^X/G/1$) at the three epochs of times under consideration, but also discuss some other interesting properties of $M^X/G/1$ and unify the results of many authors. As Foster did, so shall we use the term 'queue size' in all the three cases under discussion.

The symbol E_R will indicate a modified Erlangian distribution with p.d.f.

$$\sum_{r=1}^{\infty} c_r\{(\mu t)^{r-1}/(r-1)!\}\mu \exp(-\mu t).$$

E_k which can be obtained from E_R by putting $c_r = \delta_{rk}$, where δ_{rk} is a Kronecker symbol, will then be the usual Erlang (gamma) distribution which is the convolution of k exponential distributions with means $1/\mu$.

THE SYSTEM $M^X/G/1$

P(z)

Let $N(t)$ be a random variable (r.v.) representing queue size at time t. Let groups of customers arrive at epochs $0 = \sigma_0', \sigma_1', \ldots, \sigma_n', \ldots$ with their size being a r.v., X, such that

$$P(X = x) = a_x, \quad \sum_i a_x = 1 \quad \text{and} \quad \bar{a} = \sum^{\infty} xa_x < \infty.$$

The arrival epochs follow Poisson (random) distribution with mean $1/\lambda$. The service times of individual customers are independently distributed identical r.v.'s with common density $b(v)$ such that $1/\mu = \int_0^{\infty} vb(v)dv < \infty$. Let $\sigma_1, \sigma_2, \ldots, \sigma_n, \ldots$ be the epochs of departures of customers from the system.

Let us now define the following probabilities:

(i) $P_j = \lim_{t\to\infty} P(N(t) = j)$

This means P_j is the limiting probability (as $t \to \infty$) of j in the system at a random epoch of time.

(ii) $P_j^- = \lim_{n\to\infty} P(N(\sigma_n' - 0) = j)$

This means P_j^- is the limiting probability (as $n \to \infty$) of j in the system just before an arrival epoch.

(iii) $P_j^+ = \lim_{n\to\infty} P(N(\sigma_n + 0) = j)$

This means P_j^+ is the limiting probability (as $n \to \infty$) of j in the system just after a departure epoch.

Assuming that the various limiting probabilities exist which they do when $\rho = \lambda\bar{a}/\mu < 1$, it will be established that

(1) $$P^{-}(z) = P(z) = \{\bar{a}(1 - z)/[1 - A(z)]\}P^{+}(z),$$

where

$$P(z) = \sum_{0}^{\infty} P_j z^j, \text{ etc.}$$

are the p.g.f.'s.

The first equation of (1) is easily established because of the randomness of arrivals. To prove the second equation requires a bit of more rigorous argument. We first discuss $P(z)$ and then its relation to $P^{+}(z)$. To get $P(z)$, we introduce the following notation. Let

(a) $\eta(x)$ be the conditional service rate so that the service time density and distribution functions are, respectively, given by

$$b(x) = \eta(x) \exp\left[-\int_0^x \eta(t)\, dt\right]$$

$$B(x) = 1 - \exp\left[-\int_0^x \eta(t)\, dt\right]$$

(b) $P_n(x, t) = \lim_{\Delta x \to 0} \{P\{N(t) = n, x < X(t) > x + dx\}/\Delta x\}$
where $X(t)$ is the elapsed service time of the customer undergoing service at time t.

(c) $P_0(t) = P[N(t) = 0]$

Now to find the p.g.f. $P(z)$ of $N(t)$ in the limiting case, we proceed as in Cox [2]. Since the arguments and steps in deriving the steady-state equations are the same, we only give the partial differential equations in the limiting case as $t \to \infty$ with the notation

$$\lim_{t\to\infty} P_n(x, t) = P_n(x), \quad \lim_{t\to\infty} P_0(t) = P_0;$$

(2) $$0 = -\lambda P_0 + \int_0^{\infty} P_1(x)\eta(x)\, dx,$$

(3) $$\frac{\partial P_n(x)}{\partial x} = -(\lambda + \eta(x))P_n(x) + \sum_{m=1}^{n} a_m P_{n-m}(x), \quad n \geqslant 1$$

which are to be solved under the so-called boundary condition

(4) $$P_n(0) = \int_0^{\infty} P_{n+1}(x)\eta(x)\, dx + \lambda a_n P_0, \quad n \geqslant 1$$

and the normalizing condition

(5) $$P_0 + \sum_{n=1}^{\infty} \int_0^{\infty} P_n(x)\, dx = 1.$$

Define the p.g.f.'s

(6) $$P_0(z; x) = \sum_{n=1}^{\infty} P_n(x) z^n, \quad A(z) = \sum_{m=1}^{\infty} a_m z^m$$

Once again applying Cox's procedure to equations from (2) and (3) and using (4) and (6), we get the following results:

(7) $$P_0(z; x) = P_0(z; 0)(1 - B(x)) \exp[\lambda(A(z) - 1)x]$$

and consequently

$$P_0(z) = \int_0^\infty P_0(z; x)\, dx$$
$$= zP_0\{\bar{b}(\lambda - \lambda A(z)) - 1\}/[z - \bar{b}(\lambda - \lambda A(z))],$$

where

$$\bar{b}(\alpha) = \int_0^\infty e^{-\alpha x} b(x)\, dx,$$

and

$$P_0(z; 0) = \lambda z P_0[A(Z) - 1]/[z - \bar{b}(\lambda - \lambda A(z))],$$

which can be obtained by using (4) and (7).

Finally,

(8)
$$P(z) = P_0(z) + P_0 = P_0(1 - z)/[1 - \{z/\bar{b}(\lambda - \lambda A(z))\}],$$

where

$$P_0 = 1 - \rho.$$

The result given in (8) for the case when $G = E_R$ or E_k has been independently discussed by Gupta [8] and Restrepo [13] respectively. Gaver [6] obtained (8) by using renewal theoretic arguments. We can, as well, obtain (8) if we identify a group with a single customer so that its (group's) service time distribution is just the total service time of its members constituting the group and then use the results of a single server system M/G/1. However, as this has been pointed out earlier that the present approach is different, and it not only immediately leads to the result for $P^+(z)$, but also unifies all the results reported so far, besides correcting an error in Krakowski's [11] results.

$P^+(z)$

To get $P^+(z)$ or relate $P(z)$ to $P^+(z)$, we first find $P^+(z)$ and then the relation is easily established. To get $P^+(z)$, we have

$$P_n^+ = D \int_0^\infty P_{n+1}(x)\eta(x)\, dx,$$

where D is a normalizing constant. The p.g.f. $P^+(z)$ is, then, given by

$$P^+(z) \equiv \sum_{n=0}^{\infty} P_n^+ z^n$$
$$= D \int_0^\infty \sum_{n=1}^{\infty} z^n P_{n+1}(x)\eta(x)\, dx$$
$$= (D/z) P_0(z; 0) \int_0^\infty b(x) e^{-\lambda[1 - A(z)]x}\, dx$$

(9)
$$= \lambda D P_0(A(z) - 1)/[\{z/\bar{b}(\lambda - \lambda A(z))\} - 1],$$

where we have used $P(z; 0)$. Using the normalizing condition and the value of $P(z)$, we get

$$P^+(z) = \frac{1 - A(z)}{(1 - z)\bar{a}} P(z)$$

as stated in (1). If $a_r = \delta_{r1}$, then $P^+(z) = P(z)$, a result first established by Khintchine [10] and later by other authors. It can be shown that the result (9) agrees with the one obtained by Sahbazov [13] using the imbedded Markov chain procedure. Now we wish to discuss other interesting features of the system under consideration.

I. The system $M^X/M/1$. This may be obtained from $M^X/G/1$ by putting $G = E$ so that if $\eta(x) = \mu$, then (8) gives (since $\bar{b}(a) = \mu/(\mu + \alpha)$)

$$P(z) = (1 - \rho)(1 - z)\mu/[\mu - z\{\mu + \lambda - \lambda A(z)\}], \tag{10}$$

which is Luchak's [12] result for $M/E_X/1$. It is thus interesting to see that the system $M^X/M/1$ and $M/E_X/1$ are equivalent, not only when $P(X = r) = 1$ (see, e.g., later part of section 4.3.1 of Gross and Harris [7]), but also when X is a r.v. The equivalence of the more general systems $GI^r/M/1$ and $GI/E_r/1$ wherein r is fixed has been considered by several authors, see, for example Foster [4]. It is possible, in principle, to show that $GI^X/M/1$ and $GI/E_X/1$ are equivalent, but the analysis would be a bit more cumbersome.

II. From the relation (1), one can easily see that

$$\bar{a}P_0^+ = P_0 \tag{11}$$

The result (11) is new and exhibits an interesting phenomenon. It states that an observer is more likely to find the system empty (for $\bar{a} > 1$) than a departing customer leaves it. Its accuracy can easily be checked. For when $\bar{a} = 1$, (11) reduces to the known relation $P_0 = P_0^+$ for the single server queueing system M/G/1.

III. Another interesting case which is connected with the case II or equation (11) is the relation between the imbedded Markov chain probabilities P_0^+ and π_0. From Gross and Harris [7] or Harris [9], one can find that

$$\pi_0 = \frac{1 - \rho}{1 - \rho + \bar{a}} = \frac{P_0}{P_0 + \bar{a}} = \frac{P_0^+}{P_0^+ + 1}, \tag{12}$$

where the last result has been obtained by using the equation (11).

IV. The various moments of queue size may be obtained from (1). In particular, if L^-, L, L^+ denote the expected queue sizes at the three epochs of time—just before arrival, random and just after departure, then one can easily see from (1) that the following relations must be satisfied.

$$L^- = L = L^+ - \frac{\sigma_a^2 + (\bar{a})^2 - \bar{a}}{2} \tag{13}$$

where one may obtain L from (8) and is given by

$$L = \rho + \frac{\rho^2(\mu^2\sigma_s^2 + 1)}{2(1 - \rho)} + \rho\,\frac{\sigma_a^2 + (\bar{a})^2 - \bar{a}}{2(1 - \rho)\bar{a}} \tag{14}$$

where

σ_s^2 = *variance of the service time distribution, and*
σ_a^2 = *variance of the group size distribution.*

One can see from (14) that the average number in the queue, L_q, is given by

$$L_q \equiv L - \rho = \frac{\lambda\bar{a}\rho R}{1 - \rho} + \rho\,\frac{\sigma_a^2 + (\bar{a})^2 - \bar{a}}{2(1 - \rho)\bar{a}} \tag{15}$$

where $2\mu R = (\mu^2\sigma_s^2 + 1)$ and R is the same as defined by Krakowski [11]. L_q is now the more general and correct form of Krakowski's [11] result (S.23) which is true only when the group size is constant rather than a random variable. Once L_q is known, one can obtain W_q

from Little's formula, $L_q = \lambda \bar{a} W_q$. Equation (13) shows that an observer (for $\bar{a} > 1$) is more likely to see a shorter expected queue size than a departing customer leaves it, which is consistent with the remark made in case II.

V. If one is interested in the distribution of the number in the queue, it may be obtained from $P(z)$. For, if one defines, in the case when $t \to \infty$, N_q as a r.v. for the number in the queue, then as

$$N_q = \begin{cases} N - 1, & N \geqslant 2 \\ 0 \quad , & N \leqslant 1 \end{cases}$$

$$P_q(z) = E[z^{N_q}] = P_0(z - 1)/[z - \bar{b}(\lambda - \lambda A(z)].$$

This has an interesting interpretation. It shows that the p.g.f. of the number in the system at a random epoch, for the bulk arrival system $M^X/G/1$, is equal to the p.g.f. of the number in the queue times the p.g.f. of the number that arrive during the service time of a customer. Such an interpretation for the sytem M/G/1 where-in arrivals are by singlets is well known.

VI. An interesting result which falls outside the preceding results is the expected busy-period of the server. One way to find the expected busy-period of the server is to first find the distribution of busy-period, and then from it the expected value. However, its derivation by using an alternating renewal process is elegant. It is this approach that we adopt here. Since idle-periods and busy-periods generate an alternating renewal process, we have from the theory of renewal processes,

$$E(X)/E(Y) = \rho/(1 - \rho),$$

where $E(X)$ and $E(Y)$ are the expected busy and idle periods respectively. Now, since in $M^X/G/1$ by using the forgetfulness property of the exponential,

$$E(Y) = 1/\lambda, \quad E(X) = \bar{a}/(\mu - \lambda \bar{a})$$

which reduces to the well-known result for the queueing system M/G/1 if we take

$$a_r = \delta_{r1}.$$

ACKNOWLEDGMENT

The research for this paper was supported (in part) by the Defense Research Board of Canada, Grant Number 3610-603. The author is extremely grateful to a referee for pointing out the relation (12) and a few other useful recommendations.

REFERENCES

[1] Burke, P.J., "Delays in Single-Server Queues with Batch Input," Operations Research 23, 830-833, 1975.

[2] Chaudhry, M.L. and J.G.C. Templeton, "The Queueing System $M/G^B/1$ and its Ramifications," Under submission.

[3] Cox, D.R., "The Analysis of Non-Markovian Stochastic Processes by the Inclusion of Supplementary Variables," Proceedings of the Cambridge Philosophical Society 51, 433-441, 1955.

[4] Foster, F.G., "Batched Queueing Processes," Operations Research 12, 441-449, 1964.

[5] Foster, F.G. and A.G.A.D. Perera, "Queues with Batch Arrivals II," Acta Mathematica Academiae Scientiarum Hungaricae, 16, 275-287, 1965.

[6] Gaver, D.P., "Imbedded Markov Chain Analysis of a Waiting Line Process in Continuous Time, Annals of Mathematical Statistics 30, 698-720, 1959.
[7] Gross, D. and C.M. Harris, Fundamentals of queueing theory (John Wiley and sons, 1974).
[8] Gupta, S.K., "Queues with Batch Poisson Arrivals and a General Class of Service Time Distributions," Journal of Industrial Engineering 15, 319-320, 1964.
[9] Harris, C.M., "Some Results for Bulk-Arrival Queues with State Dependent Service Times," Management Science 16, 313-326, 1970.
[10] Khintchine, A., "Mathematical Theory of a Stationary Queue," Matemateceskii Sbornik, 39, 73-84 (Russian), 1932.
[11] Krakowski, M., "Arrival and Departure Processes in Queues," Revue Francaise Automatique Informatique et Recherche Opérationelle V-1, 45-56, 1974.
[12] Luchak, G., "The Continuous Time Solution of the Equations of the Single Channel Queue with a General Class of Service Time Distributions by the Method of Generating Functions," Journal of Royal Statistics Society Service B20, 176-181, 1958.
[13] Restrepo, R.A., "A Queue with Simultaneous Arrivals and Erlang Service Distribution," Operations Research 13, 375-381, 1965.
[14] Sahbazov, A.A., "A Problem of Service with Non-Ordinary Demand Flow," Soviet Mathematics Doklady 3, 1000-1003, 1962.

ON THE MOMENTS OF GAMMA ORDER STATISTICS

P. C. Joshi

Department of Mathematics
Indian Institute of Technology
Kanpur, India

ABSTRACT

A recurrence relation between the moments of order statistics from the gamma distribution having an integer parameter r is obtained. It is shown that if the negative moments of orders $-(r-1), \ldots, -1$ of the smallest order statistic in random samples of size n are known, then one can obtain all the moments. Tables of negative moments for $r = 2\ (1)\ 5$ are also given.

1. INTRODUCTION

Let X be a gamma random variable with probability density function

$$f(x) = e^{-x}x^{r-1}/\Gamma(r), \quad x > 0, \tag{1}$$

where $r > 0$. Let $X_1, X_2, \ldots, X_n$ be a random sample from (1), and $X_{1,n} \leqslant X_{2,n} \leqslant \ldots \leqslant X_{n,n}$ be the corresponding order statistics. Denote the i th moment of $X_{k,n}$ by $\alpha_{k,n}^{(i)}$.

An expression for $\alpha_{k,n}^{(i)}$ is given by Gupta [4] when r is an integer, and by Krishnaiah and Rizvi [6] for a general value of r. Tables of moments for selected values of n, k, r and i are given by Breiter and Krishnaiah [2] and Gupta [4]. The gamma order statistics and their moments are of great use in the analysis of life testing data, especially for $r = 1$ when the gamma distribution reduces to the exponential distribution. Some applications of gamma order statistics are discussed in Gupta [4] and Young [8].

The moments of order statistics are known to satisfy some recurrence relation, for example, see David ([3], pp. 36-38). In particular

$$\alpha_{k,n}^{(i)} = \sum_{j=n-k+1}^{n} \binom{j-1}{n-k} \binom{n}{j} (-1)^{j-n+k-1}\, \alpha_{1,j}^{(i)}\,. \tag{2}$$

Thus the moments of $X_{k,n}$ can be obtained as a linear combination of moments of smallest order statistics in random samples of $n - k + 1, \ldots, n$. In this paper, we derive another type of recurrence relation when r is an integer. In fact we show that higher order moments of $X_{k,n}$ can be obtained from those of the lower order. Recurrence relations of this type for specific distributions are given by Barnett [1] for Cauchy distribution, by the author [5] for the exponential and truncated exponential distributions, and by Shah [7] for logistic distribution. In addition we provide a table of negative moments $\alpha_{k,n}^{(i)}$ for $r = 2\ (1)\ 5$ and $i = -(r-1), \ldots, -1$.

2. THE NEGATIVE MOMENTS

For the gamma random variable X with density given by (1), the i th moment

$$E(X^i) = \Gamma(r+i) / \Gamma(r)$$

exists for all $i > -r$. Consequently, the i th moment of $X_{k,n}$ also exists for $i > -r$ (David [3], pp. 25-26). When r is a positive integer, then

$$\alpha_{k,n}^{(i)} = \frac{n!}{(k-1)!\,(n-k)!} \int_o^\infty x^i \{F(x)\}^{k-1}\{1-F(x)\}^{n-k} f(x)\, dx,$$

where $F(x)$ is the cumulative distribution function of X given by

$$F(x) = 1 - \sum_{j=0}^{r-1} e^{-x} x^j/j!, \quad x > 0.$$

Gupta [4] has shown that this can be written as

$$\alpha_{k,n}^{(i)} = \frac{n!}{(k-1)!\,(n-k)!\,\Gamma(r)} \sum_{p=0}^{k-1} (-1)^p \binom{k-1}{p}$$

$$\sum_{m=0}^{(r-1)(n-k+p)} a_m(r,\ n-k+p)\ \frac{\Gamma(r+i+m)}{(n-k+p+1)^{r+i+m}},$$

where $a_m(r,\ p)$ is the coefficient of t^m in the expansion of $\left[\sum_{j=0}^{r-1} t^j/j!\right]^p$. For $r = 1\ (1)\ 5$, he has used this relation for tabulating the first four moments of $X_{k,n}$ for $1 \leqslant k \leqslant n \leqslant 10$, and of $X_{1,n}$ for $n = 11\ (1)\ 15$. In Table 1, we extend his tables to negative moments $\alpha_{k,n}^{(i)}$ for $r = 2\ (1)\ 5$, $i = -(r-1), \ldots, -1$ and $1 \leqslant k \leqslant n \leqslant 10$, and $\alpha_{1,n}^{(i)}$ for $n = 11\ (1)\ 25$ and same values of r and i. These were evaluated to eight significant digits and are correct to the five decimal places as tabulated. For $n \leqslant 10$, they were also checked by using the identity

$$\sum_{k=1}^{n} \alpha_{k,n}^{(i)} = n\ \alpha_{1,1}^{(i)}.$$

3. THE RECURRENCE RELATION

In [5], the author has shown that for the exponential distribution ($r = 1$)

$$\alpha_{k,n}^{(i)} = \alpha_{k-1,n-1}^{(i)} + (i/n)\,\alpha_{k,n}^{(i-1)}, \quad i = 1, 2, \ldots; 1 \leqslant k \leqslant n,$$

where we follow the conventions

$$\alpha_{k,n}^{(0)} = 1, \qquad 1 \leqslant k \leqslant n, \tag{3}$$

$$\alpha_{0,t}^{(i)} = 0, \qquad i = 1, 2, \ldots; t = 0, 1, 2, \ldots \tag{4}$$

This recurrence relation was then extended to the right truncated exponential distribution. We now generalize this result in another direction to the gamma distribution and show that for integral values of r

$$\alpha_{k,n}^{(i)} = \alpha_{k-1,n-1}^{(i)} + \Gamma(r)\ (i/n) \sum_{t=0}^{r-1} \alpha_{k,n}^{(t+i-r)}/t! \tag{5}$$

for $i = 1, 2, \ldots, 1 \leqslant k \leqslant n$, where the conventions given at (3) and (4) are followed. To this end, let

$$h(x) = -\left[1 - \sum_{j=0}^{r-1} e^{-x} x^j/j!\right]^{k-1} \left[\sum_{j=0}^{r-1} e^{-x} x^j/j!\right]^{n-k+1},$$

then

$$h'(x) = \left[1 - \sum_{j=0}^{r-1} e^{-x} x^j/j!\right]^{k-2} \left[\sum_{j=0}^{r-1} e^{-x} x^j/j!\right]^{n-k} \left\{\frac{e^{-x} x^{r-1}}{(r-1)!}\right\} \left\{n\left[1 - \sum_{j=0}^{r-1} e^{-x} x^j/j!\right] - (k-1)\right\},$$

and

$$\alpha_{k,n}^{(i)} - \alpha_{k-1,n-1}^{(i)} = \binom{n-1}{k-1} \int_o^\infty x^i h'(x)\, dx.$$

Integrating by parts, by treating x^i for differentiation and $h'(x)$ for integration, we have

$$\alpha_{k,n}^{(i)} - \alpha_{k-1,n-1}^{(i)} = -\binom{n-1}{k-1} \int_o^\infty i\, x^{i-1} h(x)\, dx$$

$$= i \binom{n-1}{k-1} \int_o^\infty x^{i-1} \left[1 - \sum_{j=0}^{r-1} e^{-x} x^j/j!\right]^{k-1} \left[\sum_{j=0}^{r-1} e^{-x} x^j/j!\right]^{n-k} \left[\sum_{t=0}^{r-1} e^{-x} x^t/t!\right] dx.$$

Taking the sum over t outside the integral sign, multiplying and dividing by $\Gamma(r)$, and integrating term by term we get

$$\alpha_{k,n}^{(i)} - \alpha_{k-1,n-1}^{(i)} = (i/n)\, \Gamma(r) \sum_{t=0}^{r-1} \alpha_{k,n}^{(t+i-r)}/t!,$$

which proves the results.

Relation (5) expresses the ith order moment of $X_{k,n}$ in terms of ith order moment of $X_{k-1,n-1}$ and lower order moments of $X_{k,n}$. In particular, it gives the mean of $X_{1,n}$ in terms of moments of orders $-(r-1), \ldots, -1$ of $X_{1,n}$, the second moment of $X_{1,n}$ in terms of moments of orders $-(r-2), \ldots, -1, 0, 1$ of $X_{1,n}$, etc. Taken together with relation (2), it shows that if the negative moments of orders $-(r-1), \ldots, -1$ of the smallest order statistic in samples of size $j \leqslant n$ are known, then one can calculate all the moments $\alpha_{k,n}^{(i)}$ for $i = 1, 2, \ldots$ and $1 \leqslant k \leqslant n$.

It should be noted that only non-negative terms are added for the evaluation of $\alpha_{k,n}^{(i)}$ in equation (5). Consequently, the rounding errors are negligible for small values of r. This is illustrated in the following example.

EXAMPLE: $r = 2$ and $k = 1$. In this case equation (5) reduces to

$$\alpha_{1,n}^{(i)} = (i/n)\, (\alpha_{1,n}^{(i-2)} + \alpha_{1,n}^{(i-1)}), \quad i = 1, 2, \ldots. \tag{6}$$

Thus for $n = 10$, say, we have from Table 1, $\alpha_{1,10}^{(-1)} = 3.66022$. Equation (6), then gives $\alpha_{1,10}^{(1)} = 0.46602$, $\alpha_{1,10}^{(2)} = 0.29320$, $\alpha_{1,10}^{(3)} = 0.22777$, $\alpha_{1,10}^{(4)} = 0.20839$ etc. These values agree perfectly with the values evaluated directly (see also Gupta [4]).

For all values of n, r and k, the moments of order i, $1 \leqslant i \leqslant 4$, obtained by direct evaluation and by recurrence relation (5) agree up to eight significant digits, the digits up to which the calculations were performed and on which Table 1 is based.

TABLE 1. *Table of Negative Moments $E(X_{k,n}^{i})$ of Gamma Order Statistics for $r = 2\ (1)\ 5$.*

n	k	r=2	r=3		r=4			r=5			
		i=−1	i=−2	i=−1	i=−3	i=−2	i=−1	i=−4	i=−3	i=−2	i=−1
1	1	1.00000	0.50000	0.50000	0.16667	0.16667	0.33333	0.04167	0.04167	0.08333	0.25000
2	1	1.50000	0.87500	0.68750	0.31250	0.27083	0.43750	0.08073	0.07422	0.12891	0.31836
2	2	0.50000	0.12500	0.31250	0.02083	0.06250	0.22917	0.00260	0.00911	0.03776	0.18164
3	1	1.88889	1.20370	0.82099	0.44833	0.35566	0.50810	0.11832	0.10295	0.16424	0.36321
3	2	0.72222	0.21759	0.42052	0.04084	0.10118	0.29629	0.00555	0.01676	0.05824	0.22866
3	3	0.38889	0.07870	0.25849	0.01083	0.04316	0.19560	0.00113	0.00529	0.02752	0.15813
4	1	2.21875	1.50415	0.92792	0.57746	0.42950	0.56292	0.15485	0.12928	0.19403	0.39733
4	2	0.89931	0.30236	0.50020	0.06093	0.13415	0.34365	0.00872	0.02397	0.07488	0.26085
4	3	0.54514	0.13282	0.34085	0.02074	0.06820	0.24894	0.00238	0.00954	0.04159	0.19646
4	4	0.33681	0.06066	0.23103	0.00753	0.03482	0.17783	0.00071	0.00388	0.02283	0.14536
5	1	2.51040	1.78463	1.01852	0.70158	0.49594	0.60831	0.19056	0.15387	0.22020	0.42517
5	2	1.05215	0.38222	0.56550	0.08102	0.16375	0.38135	0.01202	0.03089	0.08935	0.28598
5	3	0.67004	0.18258	0.40225	0.03081	0.08975	0.28710	0.00375	0.01360	0.05317	0.22314
5	4	0.46187	0.09965	0.29992	0.01403	0.05383	0.22349	0.00147	0.00683	0.03387	0.17868
5	5	0.30554	0.05092	0.21381	0.00590	0.03006	0.16641	0.00053	0.00314	0.02007	0.13703
6	1	2.77469	2.04988	1.09789	0.82168	0.55693	0.64736	0.22559	0.17713	0.24377	0.44883
6	2	1.18894	0.45840	0.62168	0.10103	0.19097	0.41309	0.01543	0.03757	0.10235	0.30684
6	3	0.77856	0.22986	0.45312	0.04100	0.10931	0.31788	0.00521	0.01754	0.06337	0.24427
6	4	0.56151	0.13531	0.35138	0.02061	0.07019	0.25633	0.00229	0.00965	0.04297	0.20202
6	5	0.41205	0.08182	0.27419	0.01074	0.04566	0.20707	0.00106	0.00542	0.02932	0.16701
6	6	0.28424	0.04474	0 20174	0.00493	0.02694	0.15828	0.00042	0.00269	0.01822	0.13103
7	1	3.01814	2.30292	1.16897	0.93848	0.61369	0.68180	0.26003	0.19932	0.26535	0.46951
7	2	1.31401	0.53162	0.67144	0.12093	0.21637	0.44071	0.01891	0.04403	0.11424	0.32479
7	3	0.87628	0.27533	0.49728	0.05127	0.12748	0.34404	0.00674	0.02140	0.07262	0.26198
7	4	0.64827	0.16924	0.39424	0.02730	0.08509	0.28299	0.00318	0.01240	0.05103	0.22065
7	5	0.49645	0.10986	0.31923	0.01560	0.05901	0.23633	0.00163	0.00758	0.03693	0.18805
7	6	0.37829	0.07060	0.25617	0.00880	0.04032	0.19537	0.00083	0.00455	0.02628	0.15859
7	7	0.26856	0.04043	0 19266	0.00429	0.02471	0.15210	0.00035	0.00238	0.01688	0.12644
8	1	3.24502	2.54586	1.23362	1.05244	0.66703	0.71273	0.29397	0.22060	0.28537	0.48792
8	2	1.42998	0.60239	0.71637	0.14071	0.24031	0.46528	0.02244	0.05032	0.12526	0.34060
8	3	0.96608	0.31934	0.53667	0.06159	0.14457	0.36699	0.00832	0.02517	0.08116	0.27735
8	4	0.72662	0.20197	0.43165	0.03407	0.09900	0.30580	0.00411	0.01511	0.05839	0.23637
8	5	0.56992	0.13650	0.35684	0.02053	0.07118	0.26018	0.00225	0.00970	0.04368	0.20492
8	6	0.45236	0.09387	0.29666	0.01265	0.05170	0.22203	0.00126	0.00632	0.03288	0.17793
8	7	0.35360	0.06285	0.24267	0.00752	0.03653	0.18648	0.00068	0.00397	0.02408	0.15214
8	8	0.25641	0.03723	0.18552	0.00382	0.02303	0.14718	0.00031	0.00215	0.01585	0.12277
9	1	3.45832	2.78021	1.29314	1.16395	0.71753	0.74089	0.32747	0.24112	0.30409	0.50457
9	2	1.53864	0.67103	0.75749	0.16036	0.26303	0.48749	0.02601	0.05645	0.13558	0.35478
9	3	1.04970	0.36212	0.57242	0.07194	0.16078	0.38753	0.00994	0.02887	0.08914	0.29097
9	4	0.79884	0.23377	0.46516	0.04090	0.11214	0.32591	0.00507	0.01777	0.06521	0.25009
9	5	0.63636	0.16223	0.38976	0.02553	0.08257	0.28066	0.00290	0.01178	0.04986	0.21923
9	6	0.51677	0.11592	0.33050	0.01653	0.06207	0.24379	0.00173	0.00803	0.03874	0.19347
9	7	0.42016	0.08284	0.27974	0.01070	0.04651	0.21114	0.00103	0.00546	0.02995	0.17016
9	8	0.33459	0.05714	0.23208	0.00661	0.03367	0.17943	0.00058	0.00354	0.02241	0.14699
9	9	0.24664	0.03474	0.17970	0.00348	0.02170	0.14315	0.00027	0.00197	0.01503	0.11974

TABLE 1 (Continued). *Table of Negative Moments $E(X_{k,n}^{i})$ of Gamma Order Statistics for r = 2 (1) 5.*

n	k	r = 2	r = 3		r = 4			r = 5			
		i = −1	i = −2	i = −1	i = −3	i = −2	i = −1	i = −4	i = −3	i = −2	i = −1
10	1	3.66022	3.00714	1.34843	1.27330	0.76562	0.76678	0.36057	0.26097	0.32173	0.51978
10	2	1.64122	0.73783	0.79554	0.17988	0.28472	0.50782	0.02961	0.06243	0.14532	0.36767
10	3	1.12832	0.40384	0.60530	0.08229	0.17626	0.40619	0.01160	0.03251	0.09665	0.30325
10	4	0.86626	0.26478	0.49570	0.04778	0.12466	0.34399	0.00607	0.02039	0.07161	0.26232
10	5	0.69769	0.18726	0.41935	0.03058	0.09336	0.29879	0.00357	0.01383	0.05561	0.23176
10	6	0.57502	0.13720	0.36018	0.02047	0.07178	0.26254	0.00222	0.00973	0.04411	0.20670
10	7	0.47793	0.10173	0.31072	0.01391	0.05560	0.23130	0.00140	0.00691	0.03516	0.18466
10	8	0.39541	0.07475	0.26646	0.00933	0.04262	0.20250	0.00087	0.00484	0.02772	0.16394
10	9	0.31938	0.05273	0.22349	0.00593	0.03144	0.17367	0.00051	0.00322	0.02108	0.14275
10	10	0.23856	0.03274	0.17484	0.00320	0.02061	0.13976	0.00024	0.00184	0.01436	0.11718
11	1	3.85237	3.22756	1.40017	1.38070	0.81163	0.79080	0.39330	0.28024	0.33845	0.53381
12	1	4.03607	3.44218	1.44888	1.48635	0.85582	0.81323	0.42570	0.29899	0.35437	0.54684
13	1	4.21235	3.65161	1.49497	1.59041	0.89840	0.83430	0.45779	0.31727	0.36958	0.55902
14	1	4.38203	3.85633	1.53876	1.69301	0.93953	0.85418	0.48961	0.33512	0.38418	0.57047
15	1	4.54581	4.05677	1.58052	1.79426	0.97938	0.87302	0.52116	0.35258	0.39822	0.58128
16	1	4.70426	4.25327	1.62047	1.89425	1.01804	0.89094	0.55246	0.36969	0.41175	0.59152
17	1	4.85787	4.44615	1.65880	1.99308	1.05563	0.90804	0.58353	0.38646	0.42484	0.60125
18	1	5.00706	4.63566	1.69566	2.09082	1.09224	0.92439	0.61438	0.40294	0.43751	0.61054
19	1	5.15220	4.82204	1.73118	2.18754	1.12794	0.94008	0.64503	0.41912	0.44980	0.61941
20	1	5.29358	5.00550	1.76548	2.28329	1.16279	0.95515	0.67548	0.43504	0.46173	0.62792
21	1	5.43150	5.18622	1.79865	2.37812	1.19686	0.96967	0.70574	0.45071	0.47335	0.63609
22	1	5.56620	5.36436	1.83079	2.47210	1.23021	0.98368	0.73583	0.46615	0.48467	0.64395
23	1	5.69788	5.54007	1.86198	2.56525	1.26287	0.99721	0.76574	0.48136	0.49570	0.65152
24	1	5.82675	5.71349	1.89227	2.65762	1.29488	1.01031	0.79550	0.49636	0.50647	0.65884
25	1	5.95298	5.88473	1.92174	2.74924	1.32630	1.02300	0.82510	0.51117	0.51700	0.66591

ACKNOWLEDGMENT

The author wishes to thank the referee for some helpful suggestions in the preparation of his paper.

REFERENCES

[1] Barnett, V.D., "Order Statistics Estimators of the Location of the Cauchy Distribution," Journal of American Statistical Association *61* 1205-18 (1966). Correction *63*, 383-5 (1968).

[2] Breiter, M.C. and P.R. Krishnaiah, "Tables for the Moments of Gamma Order Statistics," *Sankhya B 30*, 59-72 (1968).

[3] David, H.A., "Order Statistics," (Wiley: New York 1970).

[4] Gupta, S.S., "Order Statistics from the Gamma Distribution," Technometrics *2*, 243-62 (1960).

[5] Joshi, P.C., "Recurrence Relations Between Moments of Order Statistics from Exponential and Truncated Exponential Distributions," Sankhyā *B 39*, 362-71 (1978).

[6] Krishnaiah, P.R. and M.H. Rizvi, "A Note on the Moments of Gamma Order Statistics," Technometrics *9*, 315-8 (1967).

[7] Shah, B.K., "Note on the Moments of a Logistic Order Statistics," *Annals of Mathematical Statistics 41*, 2150-2 (1970).

[8] Young, D.H., "Moment Relations for Order Statistics of the Standardized Gamma Distribution and the Inverse Multinomial Distribution," Biometrika *58*, 637-40 (1971).

A NEW STORAGE REDUCTION TECHNIQUE FOR THE SOLUTION OF THE GROUP PROBLEM

Richard V. Helgason and Jeff L. Kennington

Department of Operations Research
and Engineering Management
Southern Methodist University
Dallas, Texas

ABSTRACT

This paper shows that by making use of an unusual property of the decision table associated with the dynamic programming solution to the goup problem, it is possible to dispense with table storage as such, and instead overlay values for both the objective and history functions. Furthermore, this storage reduction is accomplished with no loss in computational efficiency. An algorithm is presented which makes use of this technique and incorporates various additional efficiencies. The reduction in storage achieved for problems from the literature is shown.

I. INTRODUCTION

The group theoretic approach to integer programming was first presented by Ralph Gomory [4] in 1969. The basic theoretical results may be found in [3, 4, 7, 9, 10, 11] and computational experience with variations of the approach may be found in [2, 5, 6]. The first step is to solve the continuous relaxation of the integer program. If the solution is integer, the problem is solved. If not, one then uses the optimal linear programming basis to derive a relaxation of the integer program known as the group (knapsack) problem. Mathematically, the group problem may be assumed to take the following form:

$$\min \quad \sum_{i=1}^{i=q} c_i x_i$$

$$(1) \qquad s.t. \quad \sum_{i=1}^{i=q} g_i x_i \equiv d \pmod{\epsilon}$$

$$x_i \text{ a non-negative integer for all } i,$$

where $g_1, \ldots, g_q$, d, and ϵ are known integer r-vectors and the c_i's are known non-negative scalars. The details for obtaining the group problem in the above form may be found in [3, 7, 9]. The integer vectors g_i may be used to generate an abelian group under addition modulo ϵ and hence the names *group theoretic approach* and *group problem.*

The next step in the group theoretic approach is to solve (1). Gomory [4] presents a simple dynamic programming algorithm for solving this problem. Dynamic programming (see [1, 8]) is a multi-stage solution procedure in which a recursive relation is used to compute columns

in a decision table. Each successive column represents a further stage in the optimization procedure. A group problem with q columns will require a q-stage optimization. At the conclusion of the q^{th} stage, the solution may be recovered by backtracking through the decision table.

In this paper, we present an unusual property of this particular decision table which allows one to recover the optimal solution from the information associated with only the q^{th} stage. Consequently, we can dispense with table storage as such for the dynamic programming technique by overlaying all table values at successive stages.

II. SOLVING GROUP PROBLEMS VIA DYNAMIC PROGRAMMING

Consider the following two-parameter class of group minimization problems over the group $G = \{g_0, g_1, \ldots, g_{m-1}\}$,

$$P_{lk}\begin{cases} \min & \sum_{i=1}^{i=k} c_i x_i \\ s.t. & \sum_{i=1}^{i=k} \mathfrak{g}_i x_i \equiv g_l \pmod{\epsilon} \\ & x_i, \text{ a non-negative integer for all } i \end{cases}$$

where the integers l and k are bounded by $0 \leqslant l \leqslant m-1$ and $1 \leqslant k \leqslant q$. $g_0 = \underline{0}$, the zero vector. Let f_{lk} denote the optimal objective value of P_{lk} if a solution exists and let $f_{lk} = \infty$, otherwise. f_{00} is taken to be 0 while $f_{l0} = \infty$ for all $l \geqslant 1$. At the k^{th} stage of the dynamic programming solution procedure, one must find f_{rk} for $r = 0, 1, \ldots, m-1$; that is, (1) is solved using only $x_1, \ldots, x_k$ and with all group elements as right-hand sides of the congruence. If l^* is such that $g_{l^*} = d$, then the solution of P_{l^*q} is also the solution of (1).

A dynamic programming algorithm can be developed for P_{lk} by noting that the solutions to P_{lk} can be partitioned into those with $x_k = 0$ and those with $x_k \geqslant 1$. Let l' be defined such that $g_{l'} = g_l - \mathfrak{g}_k$. Then the above is equivalent to saying that for $x_k = 0$, $f_{lk} = f_{l,k-1}$; and for $x_k \geqslant 1$, $f_{lk} = c_k + f_{l'k}$. Thus a recursive equation may be written as follows:

$$f_{lk} = \min \{f_{l,k-1},\ c_k + f_{l'k}\}. \tag{2}$$

To apply (2), one must be able to compute f_{lk} for all l. This is always possible if $\mathfrak{g}_k$ generates the whole group. In the case where $\mathfrak{g}_k$ does not generate G, the following procedure is used. For each coset, one chooses an element $g_{l'}$ and sets $f^*_{l'k} = f_{l',k-1}$. Then one generates elements of the coset successively using $g_l = g_{l'} + \alpha\mathfrak{g}_k$, for $\alpha = 1, 2, \ldots$, and computes

$$f^*_{lk} = \min \{f_{l,k-1},\ c_k + f^*_{l^*k}\} \tag{3}$$

with l^* determined by $g_{l^*} = g_l - \mathfrak{g}_k$.

The above procedure is cyclic, and should be terminated when all new f^*_{lk} agree with those computed in the previous cycle. Then, f_{lk} is taken to be f^*_{lk}. Termination is guaranteed within two cycles and occurs anytime during the second cycle when any f^*_{lk} agrees with its previous value. Obviously this procedure may also be applied when $\mathfrak{g}_k$ generates the whole group since the coset of interest is G. Justification for the above procedure may be found in [3, 7, 9,

10]. In order to recover the solution at the termination of stage q, one also carries a history function, which keeps track of the variable used in (3) to obtain the minimum. Note that in case $f_{l,k-1} = c_k + f^*_{l \cdot k}$, an arbitrary decision is possible. This gives rise to a number of possible realizations of the history function, corresponding to the combinations of alternate optima for each problem P_{lk}. To facilitate the intended storage reduction, we define a particular history function as follows:

$$h_{lk} = \begin{cases} h_{l,k-1}, \text{ if } f_{lk} = f_{l,k-1}, \text{ and} \\ k, \text{ otherwise.} \end{cases} \tag{4}$$

Note that this records the most recent stage for which a strict decrease occurred in the objective value associated with g_l. Thus, in case $f_{l,k-1} = c_k + f^*_{l \cdot k}$, one does not use the variable associated with the new stage.

A naive implementation of the above algorithm requires a decision table with $mq(t + p)$ bits where t is the number of bits required to carry the f_{lk}'s and p must be such that $2^p \geqslant q + 1$ (i.e., with p bits we represent the numbers 0 to q). This algorithm as described in [7, 9, 10] requires the full table size while the presentation in [3] assumes a table with $mq(t + 1)$ bits. However, all values of f_{kl} and f^*_{lk} need not be saved since the recursions (2) and (3) can be executed with partial information about the current stage and partial information about the previous stage. Therefore, one may easily, implement the algorithm with a table of size $m(t + q)$ bits. Even so, the dynamic programming decision table may become quite large. We remark that with a table of size $m(t + q)$ bits it is possible to recover all alternate optima.

Implementation of the procedure may be enhanced if the group elements, $G = \{g_0, g_1, \ldots, g_{n-1}\}$, are ordered. Since $\underline{0} \leqslant g_i < \epsilon$ for all i, there is a natural ordering of the group elements. For any element of G, say $\beta = [\beta_1, \ldots, \beta_r]$, we assign the order of β, denoted $l(\beta)$, as follows:

$$l(\beta) = \sum_{i=2}^{i=r} \left(\prod_{k=1}^{k=i-1} \epsilon_k \right) \beta_i + \beta_1.$$

This corresponds to array subscripting, using r subscripts with the first varying most rapidly.

Using the above ordering, the recovery procedure is quite simple and is given below:

1. [Initialize Variables] $x_i \leftarrow 0$, $i = 1, \ldots, q$
2. [Start at d] $g \leftarrow d$
3. [Start at stage q] $k \leftarrow q$
4. [Reference history] $k \leftarrow h_{l(g),k}$
5. [Add to solution] $x_k \leftarrow x_k + 1$
6. [Backtrack] $g \leftarrow g - g_k$
7. [Done?] If $g \neq \underline{0}$, go to 4; otherwise, terminate.

Step 4 above implies that the history function for each group element, $G = \{g_0, \ldots, g_{m-1}\}$, and each stage, $k = 1, \ldots, q$, must be available for solution recovery.

III. DYNAMIC PROGRAMMING ALGORITHM USING THE STORAGE REDUCTION TECHNIQUE

In this section it is shown that $h_{l(g),k}$ may be replaced by $h_{l(g),q}$ in step 4 of the recovery procedure. Hence, only the q^{th} stage history function need be available for solution recovery. Consider the following propositions.

PROPOSITION 1: If $h_{lk} = j \leqslant k$, then $h_{lj} = h_{l,j+1} = \ldots = h_{lk} = j$. The proof of proposition 1 is obvious by the definition of the history function given in (4).

PROPOSITION 2: For all integers l and k with $0 \leqslant l \leqslant m-1$ and $1 \leqslant k \leqslant q$, if $j = h_{lk}$, then $h_{l'k} \leqslant j$ where $g_{l'} = g_l - \mathfrak{g}_j$.

PROOF: Choose l and k arbitrarily and let $j = h_{lk}$. Let $g_{l'}$ be such that $g_{l'} = g_l - \mathfrak{g}_j$, and let $j' = h_{l'k}$. We must show that $j' \leqslant j$. Assume the contrary. Then by Proposition 1 and the definition of the history function,

(5) $h_{lk} = \ldots = h_{lj'} = \ldots = h_{lj} = j$ with

(6) $f_{lk} = \ldots = f_{lj'} = \ldots = f_{lj} < f_{l,j-1}$; and

(7) $h_{l'k} = \ldots = h_{l'j'} \neq h_{l',j'-1}$ with

(8) $f_{l'k} = \ldots = f_{l'j'} < f_{l',j'-1} \leqslant \ldots \leqslant f_{l'j}$.

(9) (8) implies that $f_{l'j} > f_{l'j'}$.

From (6) we have that

$$f_{lj'} = f_{l'j} + c_j.$$

Let $[x_1 = \alpha_1, \ldots, x_j = \alpha_j, \ldots, x_{j'} = \alpha_{j'}]$ denote an optima for $f_{l'j'}$. Then $[x_1 = \alpha_1, \ldots, x_j = \alpha_j + 1, \ldots, x_{j'} = \alpha_{j'}]$ is feasible for $P_{lj'}$ since $g_l = g_{l'} + \mathfrak{g}_j$ and has value $f_{l'j'} + c_j$. Since the objective value for an optima of $P_{lj'}$ must be less than or equal to the objective value for any feasible solution, $f_{l'j} + c_j \leqslant f_{l'j'} + c_j$. This implies $f_{l'j} \leqslant f_{l'j'}$ which contradicts (9). Therefore $j' \leqslant j$ and the proposition is proved.

We now use the above propositions to prove that $h_{l(g),q}$ can replace $h_{l(g),j}$ in step 4 of the recovery algorithm.

PROPOSITION 3: For all integers l with $0 \leqslant l \leqslant m-1$, if $j = h_{lq}$, then $h_{l'q} = h_{l'j}$ where $g_{l'} = g_l - \mathfrak{g}_j$.

PROOF: Choose l arbitrarily and let $j = h_{lq}$. Let $g_{l'}$ be such that $g_{l'} = g_l - \mathfrak{g}_j$. By Proposition 2, $h_{l'q} \leqslant j$. Then from Proposition 1 $h_{l'q} = h_{l'j}$.

Since only the q^{th} stage history function is required for solution recovery, we drop the subscript associated with the stage for both the f_{lk}'s and h_{lk}'s. The complete dynamic programming algorithm may then be stated as follows:

REDUCED STORAGE D.P. ALGORITHM FOR THE GROUP PROBLEM

1. *Initialize*
   a. [Objective values] $f_0 \leftarrow 0$; $f_i \leftarrow \infty$, $i = 1, \ldots, m-1$.
   b. [History values] $h_i \leftarrow q+1$; $i = 0, 1, \ldots, m-1$.
   c. [Stage counter] $k \leftarrow 0$.
2. *Begin New Stage*
   a. [Increment Stage counter] $k \leftarrow k+1$.
   b. [Flag group elements not updated] $h_i \leftarrow -h_i$, $i = 0, 1, \ldots, m-1$.
   c. [Select first coset element] $g \leftarrow d + \mathfrak{g}_k$
3. *Find a Coset Element Which May Lead to an Improved Solution*
   a. [Save starting index] $i^* \leftarrow l(g)$
   b. [Test objective value] If $f_{l(g)} < f_{l(d)}$, go to 4.
   c. [Generate another coset element] $g \leftarrow g + \mathfrak{g}_k$.
   d. [Coset exhausted?] If $l(g) \neq i^*$, go to *b*.
   e. [Flag coset elements updated] $h_{l(g)} \leftarrow |h_{l(g)}|$ for all g in the coset containing g_i•; go to 5.
4. *Apply Recursion to Coset*
   a. [Starting value is previous value] $\mathrm{v} \leftarrow f_{l(g)}$.
   b. [Possible next value using stage k generator] $\mathrm{v} \leftarrow \mathrm{v} + c_k$.
   c. [Flag element updated] $h_{l(g)} \leftarrow |h_{l(g)}|$.
   d. [Generate another coset element] $g \leftarrow g + \mathfrak{g}_k$.
   e. [Test for minimum] If $f_{l(g)} \leqslant \mathrm{v}$, go to *h*.
   f. [Decrease in value] $f_{l(g)} \leftarrow \mathrm{v}$.
   g. [Update history] $h_{l(g)} \leftarrow k$; go to *b*.
   h. [Element previously updated?] If $h_{l(g)} > 0$, go to 5; otherwise, go to *a*.
5. *Test to Terminate, Go to Next Stage, or Update Another Coset*
   a. [Last stage?] If $k = q$, recover solution and terminate.
   b. [Test for another coset to update] Let g be an element for which $h_{l(g)} < 0$. If none, go to 2; otherwise go to 3.

The above algorithm is essentially equivalent to the original procedure presented by Gomory [4], except that no arbitrary decisions were possible using (3) and all information is overlaid in the decision table. Hence, our approach may save considerable core storage with no loss in efficiency. Furthermore, additional computational efficiencies have been incorporated as follows:

(i) At any stage, if no element of a particular coset has objective value less that $f_{l(d)}$, the entire coset is not actually updated since any problem solution so derived cannot be part of an optimal solution to $P_{l(d),k}$.

(ii) At each stage, the coset containing d is updated first. Thus at stage q we can terminate without updating any other cosets, and during earlier stages, a better value for $f_{l(d)}$ is used in the above test.

(iii) By using one additional bit per group element we flag cosets. Hence, no coset is ever considered more than once.

In the algorithm presented, the flag bit is the sign of h_l.

This new procedure is implemented with $m(t + p + 1)$ bits, as compared to $m(t + q)$ bits for an efficient implementation not using the overlay feature, where t denotes the number of bits required to store an f_l and p is selected such that $2^P \geq q + 2$. Table 1 presents a comparison of storage requirements on typical group problems taken from [6], with t taken as representative of the word size in bits for two classes of machines. The storage savings ranges from 12 to 85% with an average of approximately 50%.

TABLE 1. *Comparison of Table Size for Dynamic Programming Algorithm*

Table #	Problem #	Basis Determinant	# Nonbasics	$2^P \geq q + 2$	IBM 360/370 (32 bit words)			CDC 6000/7000 (60 bit words)		
(see [6])	(see [6])	m	q	p	Standard Table $m(32 + q)$ [1]	Reduced Table $m(32 + p + 1)$ [2]	% Savings [1]-[2] [1]	Standard Table $m(60 + q)$ [3]	Reduced Table $m(60 + p + 1)$ [4]	% Savings [3]-[4] [3]
3a	5	24	240	8	6528	984	85	7200	1656	77
▲	10	144	36	6	9792	5616	43	13824	9648	30
	15	180	240	8	48960	7380	85	54000	12420	77
	20	280	109	7	39480	11200	72	47320	19040	60
	25	512	140	7	69632	20480	71	83968	34816	59
▼	30	1080	109	7	152280	43200	72	182520	73440	60
3a	35	2048	104	7	278528	81920	71	335872	139264	59
3b	2	48	195	8	10896	1969	82	12240	3312	73
▲	4	128	14	4	5888	4736	20	9472	8320	12
	6	864	36	6	58752	33696	43	82944	57888	30
▼	8	5025	18	5	251250	190950	24	391950	331650	15
3b	10	6912	36	6	470016	269568	43	663552	463104	30

IV. SUMMARY

We have shown that by using a history function which records only the most recent stage at which a strict decrease occurred in the objective value associated with each group element, it is possible to dispense with table storage as such, and overlay values both for the objective and history functions. We have shown that this storage reduction may be accomplished with no loss in computational efficiency and have incorporated this technique into a highly efficient algorithm. Our procedure does not allow for the recovery of alternate optima. However, by dynamically storing a partial table consisting of all occurrences of ties in (3) following a strict decrease of f_{lk} (as given by h_{lk}), alternate optima may be determined.

REFERENCES

[1] Bellman, R.E. and S.E. Dreyfus, *Applied Dynamic Programming* (Princeton University Press, Princeton, New Jersey, 1962).

[2] Fisher, M.L., W.D. Northup and J.F. Shapiro, "Using Duality to Solve Discrete Optimization Problems: Theory and Computation Experience," Mathematical Programming Study *3*, 56-94 (1975).

[3] Garfinkel, R.S. and G.L. Nemhauser, *Integer Programming* (John Wiley and Sons, New York, New York, 1972).

[4] Gomory, R.E., "Some Polyhedra Related to Combinatorial Problems," Linear Algebra and Its Applications, *2*, 451-558 (1969).

[5] Gorry, G.A. and J.F. Shapiro, "An Adaptive Group Theoretic Algorithm for Integer Programming," Management Science, *17(5)*, 285-306 (1971).

[6] Gorry, G.A., W.D. Northrup and J.F. Shapiro, "Computational Experience With a Group Theoretic Integer Programming Algorithm," Mathematical Programming, *4*, 171-192 (1973).

[7] Hu, T.C., *Integer Programming and Network Flows* (Addison-Wesley, Reading, Mass., 1969).

[8] Nemhauser, G.L., *Introduction to Dynamic Programming* (John Wiley and Sons, New York, New York, 1966).

[9] Salkin, H.M., *Integer Programming* (Addison-Wesley Publishing Company, Reading, Mass., 1975).

[10] Taha, H.A., *Integer Programming: Theory, Applications, and Computations* (Academic Press, New York, New York, 1975).

[11] Zionts, S., *Linear and Integer Programming* (Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1974).

EXPERIMENTS WITH LINEAR FRACTIONAL PROBLEMS

Gabriel R. Bitran

Massachusetts Institute of Technology
Cambridge, Massachusetts

ABSTRACT

In this paper we present the results of a limited number of experiments with linear fractional problems. Six solution procedures were tested and the results are expressed in the number of simplex-like pivots required to solve a sample of twenty problems randomly generated.

Two main approaches emerge from the literature to solve the linear fractional problem:

(P) $$v = \max\{f(x) = n(x)/d(x): x \in F\}$$

where $n(x) = c_o + cx$, $d(x) = d_o + dx$, $F = \{x \in R^n: Ax = b\ x \geqq 0\}$, c_o and d_o are real numbers, c and d are real n-vectors, A is an mxn real matrix and b is a real m-vector. We assume in this note that F is compact and that $\min\{d(x): x \in F\} > 0$.

Charnes and Cooper [4] transform problem (P) into the linear program:

$$v = \max\{c_o t + cy: Ay - bt = 0,\ d_o t + dy = 1, \text{ and } t,\ y \geqq 0\}.$$

This approach has been extended to the nonlinear versions of (P) by Bradley and Frey [3] and Schaible [8]. The second approach solves a sequence of linear problems or at least one pivot step of each linear program over the original feasible set by updating the objective function. Algorithms in this category are related to ideas first presented by Isbell and Marlow [5] and Martos [6]. Similar algorithms have been proposed by several other authors. The interested reader is referred to the excellent bibliography collected by I.M. Stancu-Minasian [9]. Methods in the second approach propose to solve (P) through a sequence of linear programs:

$$r(x^k) = \max\{r(x^k, x) = n(x) - f(x^k)d(x): x \in F\} k = 0, 1, 2 \ldots (LP_k)$$

where x^o is a given feasible point and x^k for $k > 1$ is defined in Isbell and Marlow's procedure as being the optimal solution to (LP_{k-1}) and as the first feasible basis in (LP_{k-1}) for which $r(x^{k-1}, x) > 0$ in Martos's procedure. Both algorithms terminate at iteration k_o for which $r(x^{k_o}) = 0$. In this case $x^{k_o} = x_{\text{optimal}}$. It is worth noting that Wagner and Yuan [10] related the two main approaches by showing that Martos's algorithm is equivalent to Charnes and Cooper's method in the sense that both algorithms lead to an identical sequence of pivoting operations. Bitran and Magnanti [1] have extended the connection between these approaches by relating them to generalized programming. No theoretical or empirical evidence has been given in the past indicating which of the several existing algorithms is preferred.

In this note we present the results, in number of simplex-like pivots, of twenty problems of type (P), randomly generated, solved by the following six algorithms (each problem when solved by each of the six procedures was started with the same basic feasible solution):

A) Maximize $n(x)$ over the feasible set obtaining the optimal solution x^*. Next, apply Isbell and Marlow's algorithm with $x^o = x^*$.

B) Minimize $d(x)$ over the feasible set obtaining the optimal solution x^*. Next, apply Isbell and Marlow's algorithm with $x^o = x^*$.

C) Maximize $g(x) = [c - (cd/dd)d]x$ over F obtaining the optimal solution x^*. Next, apply Isbell and Marlow's algorithm with $x^o = x^*$ (Bitran and Novaes [2] suggested the objective function $g(x)$).

D) Isbell and Marlow's algorithm.

E) Martos's algorithm.

F) The author considered it relevant to compare these algorithms with the number of pivots necessary to solve the linear programs:

$$\text{(LP)} \qquad \max\{n(x) - \mathrm{v}d(x): x \in F\}$$

where for each of the twenty problems (P), v is chosen as its optimal value. The optimal value of (LP) is zero and any solution to (LP) is optimal in the fractional program (P) ([1]). Note that (LP) corresponds to (LP_k) with $x^k = x_{\text{optimal}}$.

The characteristics of the data of the twenty randomly generated problems are the following:

$n=40$, $m=20$, the absolute value of each a_{ij}, the (i,j)th element of each matrix A was randomly generated in the interval (0,10]. The density of negative elements being 20%. Each component b_i $i = 1, 2, \ldots, m$ of each right hand side b was defined as $\sum_{j=1}^{n} a_{ij}/2$. The objective function coefficients c_o, c_j, d_o, d_j $j = 1, 2 \ldots n$ were generated in the intervals $[-1000 \leqq c_o, c_j \leqq 1000;\ 0 < d_o, d_j \leqq 1]$, $[1 \leqq c_o, c_j \leqq 1000;\ 1 \leqq d_o, d_j \leqq 2]$ or $[-1000 \leqq c_o, c_j \leqq -1;\ 1 \leqq d_o, d_j \leqq 2]$. The reason for choosing such intervals was to obtain five problems with an angle θ between the gradients of the numerator and denominator, i.e., $\cos\theta = \dfrac{cd}{\|c\|\,\|d\|}$, in each of the four intervals $\left[0, \dfrac{\pi}{4}\right]$, $\left[\dfrac{\pi}{4}, \dfrac{\pi}{2}\right]$, $\left[\dfrac{\pi}{2}, \dfrac{3\pi}{4}\right]$, $\left[\dfrac{3\pi}{4}, 2\pi\right]$ in an attempt to identify a correlation between the algorithms tested and the geometry of linear fractional programs. The geometric properties of problem (P) are consequences of the following facts.

i) The hyperplanes $n(x) - Ld(x) = 0$ contain for each L both the sets $\{x \in R^n: f(x) = L\}$ and $CE = \{x \in R^n: n(x) = 0 \text{ and } d(x) = 0\}$. The set CE is called the center of the problem because as L varies the hyperplanes rotate about it giving a "star" centered at CE ([2]).

ii) The objective function $f(x)$ is pseudo-concave and quasiconvex on the set $\{x \in R^n: d(x) > 0\}$, i.e., $f(y) > f(x)$ if and only if $\nabla f(x)(y - x) > 0$.

In R^2 the geometry of (P) ([2]) suggests that procedure (C) would perform better than (A) and (B) for high and low values of $\theta(\theta \in [0, \pi])$. Table 1 shows the results obtained. For the first and last five problems a total of 178 pivots was necessary with procedure (C) while 233 and 363 pivots were required with procedures (A) and (B) respectively. The corresponding standard deviations being 3.70, 6.01 and 7.90. For the twenty problems selected Martos's algorithm performed better than the preceding four and in some cases required fewer pivots than procedure (F). Algorithms (C) and (D) were practically equivalent and were followed by (A), while (B) performed poorly. The computer code used to solve the twenty problems by the six algorithms was an adaptation of Burroughs's commercial code TEMPO.

TABLE 1

Problem Number	A	B	C	D	E	F	cos θ
1	24	21	13	11	12	12	.873
2	21	34	18	18	15	15	.858
3	23	34	12	12	10	10	.819
4	19	39	21	21	18	17	.770
5	32	46	22	23	21	19	.730
Mean	23.8	34.8	17.2	17.0	15.2	14.6	
Standard Deviation	4.44	8.18	4.07	4.77	3.97	3.26	
6	22	32	20	21	15	15	.569
7	25	57	28	23	18	18	.500
8	19	16	16	16	15	15	.370
9	22	51	18	20	11	11	.132
10	19	39	18	26	20	16	.076
Mean	21.4	39.0	20.0	21.2	15.8	15.0	
Standard Deviation	2.24	14.46	4.19	3.31	3.06	2.28	
11	12	38	18	18	11	10	−.103
12	21	47	21	21	19	20	−.289
13	18	33	20	22	20	21	−.424
14	18	36	20	20	19	22	−.485
15	33	50	31	25	26	19	−.613
Mean	20.4	40.8	22.0	21.2	19.0	18.4	
Standard Deviation	6.94	6.55	4.60	2.31	4.77	4.32	
16	19	51	17	17	15	15	−.720
17	16	30	13	12	13	16	−.747
18	16	39	20	21	13	15	−.820
19	33	33	22	23	21	24	−.840
20	30	36	20	24	18	19	−.874
Mean	22.8	37.8	18.4	19.4	16.0	17.8	
Standard Deviation	7.25	7.25	3.14	4.41	3.10	3.43	
Total # of Iterations	442	762	388	394	330	329	
Mean	22.1	38.1	19.4	19.7	16.5	16.4	
Standard Deviation	5.75	9.88	4.42	4.20	4.07	3.79	

TABLE 2

Algorithms compared	A B	A C	A D	A E	A F	B C	B D	B E	B F	C D	C E	C F	D E	D F	E F
W	184	110.5	69 6	182	167	190	190	210	210	10	155	188	192	162	8
σ	49.7	49.7	49.7	53.5	49.7	49.7	49.7	53.5	53.5	25.5	45.9	53.5	53.5	53.5	25.5
100 α	<.2	2.64	16.16	<.2	<.2	<.2	<.2	<.2	<.2	69.66	.2	.2	.2	.26	75.66

TABLE 3

Algorithms compared	A B	A C	A D	A E	A F	B C	B D	B E	B F	C D	C E	C F	D E	D F	E F
Null hypotheses tested	$\mu_A \geqq \mu_B$	$\mu_C \geqq \mu_A$	$\mu_D \geqq \mu_A$	$\mu_E \geqq \mu_A$	$\mu_F \geqq \mu_A$	$\mu_C \geqq \mu_B$	$\mu_D \geqq \mu_B$	$\mu_E \geqq \mu_B$	$\mu_F \geqq \mu_B$	$\mu_C \geqq \mu_D$	$\mu_E \geqq \mu_C$	$\mu_F \geqq \mu_C$	$\mu_E \geqq \mu_D$	$\mu_F \geqq \mu_D$	$\mu_F \geqq \mu_E$
W	184	110.5	69 6	182	167	190	190	210	210	10	155	188	192	162	8
σ	49.7	49.7	49 7	53.5	49 7	49.7	49 7	53.5	53.5	25.5	45.9	53.5	53.5	53.5	25.5
100 α	<.1	1.32	8.08	<.1	< 1	<.1	< 1	< 1	<.1	34.83	< 1	<.1	<.1	.13	37.83

To test if the observed differences in the number of iterations between algorithms is statistically significant, we performed Wilcoxon's signed rank test [7]. The test was used to compare the algorithms pairwise. The null hypothesis is that the distributions of the number of iterations required by the pair of algorithms being tested are identical. Table 2 shows the results obtained. The first row in the table indicates the algorithms being compared. W is the Wilcoxon statistics, σ is its standard deviation, and α is the smallest level of significance for which the null hypothesis is rejected in a two-sided symmetrical Wilcoxon's test (α represents the sum of the two tails in the test). As an example, when comparing algorithms C and E the null hypothesis is rejected for any significance level greater than .2%. The values $<.2$ in the last row of the table indicate that the value of α for the corresponding tests is smaller than .2%. The results in Table 2 suggest that the distributions of the number of iterations required by algorithms C and D and E and F are not significantly different.

A chi square test performed to test the null hypothesis that the distribution of the number of iterations for each algorithm can be approximated by a normal distribution showed that the null hypothesis cannot be rejected with a confidence level of .995. Under the assumption that the distributions of the number of iterations required by two algorithms X and Y are normal, Wilcoxon's test can be used to compare the means μ_X and μ_Y. The results of these tests are given in Table 3. W and σ are respectively the Wilcoxon statistic and its standard deviation. α, in the last row of the table, is the smallest level of significance for which the null hypothesis is rejected in a one-sided test. The null hypothesis in all tests where algorithms E and F are compared with A, B, C, and D are rejected at very low levels of significance.

REFERENCES

[1] Bitran, G.R. and T.L. Magnanti, "Duality and Sensitivity Analysis for Fractional Programs," Operations Research *24*, 675-699 (1976).

[2] Bitran, G.R. and A.G. Novaes, "Linear Programming with a Fractional Objective Function," Operations Research *21*, 22-29 (1973).

[3] Bradley, S.P. and S.C. Frey, Jr., "Fractional Programming with Homogeneous Functions," Operations Research *22*, 350-357 (1974).

[4] Charnes, A. and W.W. Cooper, "Programming with Linear Fractional Functionals," Naval Research Logistics Quarterly *9*, 181-186 (1962).

[5] Isbell, J.R. and W.R. Marlow, "Attrition Games," Naval Research Logistics Quarterly *3*, 71-93 (1956).

[6] Martos, B., "Hyperbolic Programming," Naval Research Logistics Quarterly *11*, 135-155 (1964).

[7] Mosteller, F. and R.E.K. Rourke, *Sturdy Statistics, Nonparametric and Order Statistics* (Addison-Wesley Publishing Company, Inc., Reading, MA, 1973).

[8] Schaible, S., "Parameter-Free Convex Equivalent and Dual Programs of Fractional Programming Problems," Zeitschrift für Operations Research *18*, 187-196 (1974).

[9] Stancu-Minasian, I.M., "Bibliography of Fractional Programming 1960-1976," Preprint No. 3, February 1977. Academy of Economic Studies, Depar.ment of Economic Cybernetics, Bucaresti, Romania.

[10] Wagner, H.M. and J.S.C. Yuan, "Algorithm Equivalence in Linear Fractional Programming," Management Science *14*, 301-306 (1968).

THE SENSITIVITY OF FIRST TERM NAVY REENLISTMENT TO CHANGES IN UNEMPLOYMENT AND RELATIVE WAGES*

Les Cohen

Government Services Division
Kenneth Leventhal & Company
Washington, D.C.

Diane Erickson Reedy

Mathtech, Inc.
A Division of Mathematica, Inc.
Rosslyn, Virginia

ABSTRACT

Multiple regression analysis was used to analyze newly developed twenty year time series of first term reenlistment rates for nine major Navy occupational categories. Results indicate that there are significant differences among the occupational categories in the determinants of their reenlistment behavior. More importantly, it is apparent that reenlistment rates are highly sensitive to current unemployment and especially unemployment about the time of enlistment. By comparison, relative wages (measures of military versus private sector rates of compensation) are relatively insignificant and appear powerless to control reenlistment in the context of normal fluctuations in economic activity.

I. INTRODUCTION

This paper reports the results of an analysis of first term reenlistment over the past twenty years. The study's principal objectives were to determine the uniqueness of reenlistment behavior in the Navy's major enlisted occupational categories, and in the process to measure the sensitivity of reenlistment to economic fluctuations and changes in military versus private sector rates of compensation.

In addition to the war in Viet Nam and a large number of major social, political and technological developments, the past 20 years (1958-1977) have seen highly varied economic activity. Following a long period of recovery through the Kennedy-Johnson-Viet Nam era, there have been two dramatic, successive recessions since 1969. Over the past twenty years, unemployment rates ranged from 3.5 percent to almost 9 percent, averaging 5.5 percent with a standard deviation of 1.4 percent.

Against this background of economic fluctuations, first term reenlistment rates for each of the nine major Navy occupational categories which were studied demonstrated a bimodal or saddle-shaped pattern, with a mild rise during the early 1960's and considerably more dramatic

*This research was supported by the Office of the Chief of Naval Operations, Systems Analysis Division, under contract N00074-78-C-0073 with Information Spectrum, Inc., Arlington, Virginia.

increases in the early to mid-1970's. Reenlistment rates over the twenty years, on the average, ranged between 10 and 50 percent.*

From individual monthly editions of NAVY MILITARY PERSONNEL STATISTICS, numbers of first term eligibles and reenlistments were recorded for each of the major occupational categories reported in a given month. These categories ranged from a maximum of 28 to as few as 19. The need for consistency over the twenty year sample dictated the collapsing of these groups into 17 occupational categories for which data was present throughout the time series. These were in turn combined into the nine occupational groups on which the study focused. (Apprentice categories were added to their journeyman counterparts. Precision Equipment was combined with Electronics, and Dental with Medical.)

A reduction in the number of categories was effected to increase the size of the statistics reported for each group and to minimize spurious movements in the reenlistment rates which would obscure the meaning of experimental findings. To further improve the quality of data, monthly observations were converted to quarterly, again for the purpose of increasing the number of eligibles to reasonable levels and to smooth out the time series to render real trends more readily intelligible. The resultant data base for nine Navy enlisted occupations contained reenlistment rates for 80 calendar quarters covering the twenty years from 1958 through 1977.

II. METHODOLOGY

The methodology which the study employed was multiple regression via ordinary least squares. The same equation was estimated for each of the nine occupational categories. It was decided that differences among occupations would be deduced from comparisons of individual variable performances and, to a much lesser extent, from the R^2 statistics. No attempt was made to estimate the most effective equations for each occupation. Instead, variables and transformations were selected based on their frequency of significance and general impact across all occupations evaluated collectively.

In all experiments, the dependent variable was the simple reenlistment rate, computed as in NAVY MILITARY PERSONNEL STATISTICS as the ratio of reenlistments to eligibles. Five types of independent variables were regressed against these reenlistment rates:

1. Constants, simple and seasonal

2. War Variables, constants and casualty counts representing the immediate, current period impact of the Viet Nam War

3. Motivational Variables describing the influence of the Draft and economic conditions at the time of enlistment

4. Current Economic Conditions in the Private Sector, aggregate and for occupation and industry labor market strata

5. Relative Wages, military versus private sector rates of compensation.

Plots generated for selected occupational categories revealed six observations, concentrated in the early phases of the sample, lying considerably above the normal range of values.

*All reenlistment rates studied pertained strictly to the population of recruits remaining in service for their full terms, excluding all individuals who previously separated. No consideration was given to rates of attrition or their implications for the attributes of remaining eligibles.

Upon further investigation, it was determined that these outliers were related to involuntary extensions and early separation programs, the effects of which differed noticeably among occupations and from period to period within a single occupation time series.* Rather than reducing the sample size, outliers were replaced with the average of the rates from the preceding and following quarters. While seasonal changes were not taken into account, this means of adjusting for outliers preserved the general trends of the time series.

The basic equation around which experimentation evolved was defined as follows:

$$RR = f(AUR, RW, DRAFT, WAR, S3)$$

where AUR = current unemployment rate (seasonally adjusted)

RW = ratio of military to private sector wages

DRAFT = induction levels at the time of enlistment

WAR = dummy for the Viet Nam War

S3 = third calendar quarter seasonal dummy

III. TWENTY YEAR EQUATIONS

Table 1 contains definitions for all independent variables to which reference will be made throughout the remainder of the paper. Table 2 describes equations estimated for all 80 observations. The single equation used in Table 2 is the study's baseline equation from which all findings are derived.† The baseline equation has nine independent variables, plus a tenth (DEF), for "change in definition," for Electronics and for Engineering & Hull (E&H).‡

Referring to Table 2, the statistical significance of the third quarter seasonal dummy is evident. Generally reenlistment rates are down by 3 to 5 points in the Summer and early Fall, an accounting phenomenon which results from individuals extending their terms of service for convenience into the Summer. Almost invariably, persons requesting these short-term extensions do not reenlist, thereby driving rates of reenlistment downward. The effect of these extensions is regular and of the same magnitude among the occupations.

DRFT18 represents draft levels in units of 10,000 lagged 18 periods prior to reenlistment, approximately six to nine months before enlistment.** This lag was determined experimentally, in light of uncertainty regarding the precise personal and legal points of commitment to the Navy. It is reasonable to assume that this six to nine month period is associated with deferred

*Early separation programs were instituted in 1958, 1960, and 1961. Involuntary extensions occurred in conjunction with Berlin (1961), Cuba (1962) and Viet Nam (1965).

†A review of the correlation matrix for all variables indicated no signs of multicollinearity.

‡As of the fourth calendar quarter of 1973, a change in BUPERS policy affected reenlistment rates in ratings with six year obligors. The effect of this procedural change fell primarily upon the Electronics and E&H categories which have heavy concentrations of long-term training programs. As a result, beginning in October 1973 Electronics and E&H first term reenlistment rates are artificially inflated.

To compensate, the DEF dummy variable was added to the Electronics and E&H equations for the twenty year sample. As Table 2 indicates, DEF was significant for both occupations, indicating average overstatements of reenlistment rates of 27 and 10 percentage points for Electronics and E&H respectively over the period for which DEF was in effect.

**With minor exceptions, the preponderance of Navy recruits covered by the sample enlisted for four years. Data prohibited the isolation of persons entering under three or six year programs.

TABLE 1. *Variable Definitions*

Dependent Variable		Reenlistment Rate = Ratio of Reenlistees to Eligibles (e.g., .56 = 56% reenlistment)
Independent Variables:	C	Constant
	AUR	Aggregate Seasonally Adjusted Unemployment Rate (e.g., .06 = 6% unemployment)
	ARAUR	Two Year Average Quarterly Rate of Change in AUR (e.g., -.07 = -7% average rate of decline in AUR)
	AUR13	Unemployment Rate 13 Periods Prior to Reenlistment (e.g., .06 = 6% unemployment)
	RW	Relative Wages (E-4 Base Pay to Private Sector Earnings) (e.g., .45 indicates E-4 pay is 45% of private sector wages)
	ARATE	Average Grade of Eligibles (e.g., 4.01 indicates average one point above E-4)
	WAR	Viet Nam War Dummy (1/1968 - 4/1972)
	DRFT18	Draft Levels 18 Periods Prior To Reenlistment (Inductees $\times 10^{-5}$) (e.g., .96 = 96,000 inductees)
	S3	3rd Calendar Quarter Seasonal Dummy
	DEF	Dummy For Change in Definition (e.g., 6-year obligors; Electronics and E&H only)

TABLE 2. *Total, Twenty Year Sample*
Determinants of Navy Reenlistment (Quarterly: 1/58-4/77)
Coefficients (t-Statistics) for all Variables

	Independent Variables											
	C	AUR	ARAUR	AUR13	RW	ARATE	WAR	DRFT18	S3	DEF	R^2	D-W
Deck	+0.52	+3.42	-0.35	+2.00	+0.54	−0.19	+0.02	−0.01	−0.03		.74	1.36
	(2.13)	(4.85)	(1.77)	(2.63)	(6.16)	(2.74)	(0.91)	(3.49)	(2.65)			
Ordnance	−0.36	+4.68	−0.66	+2.21	+0.69	+0.01	+0.08	−0.01	−0.05		.75	1.19
	(1.05)	(4.73)	(2.41)	(2.11)	(5.67)	(0.15)	(2.50)	(3.69)	(2.96)			
Electronics & Prec. Equip.	−1.36	+2.24	+0.33	+2.24	+0.84	+0.27	+0.17	−0.01	−0.05	+0.27	.88	1.01
	(2.76)	(1.48)	(.394)	(1.38)	(3.33)	(1.96)	(3.48)	(1.89)	(2.05)	(4.77)		
Administration	+0.45	+3.37	−0.36	+0.92	+0.26	−0.12	−0.05	−0.00	−0.05		.74	1.63
	(1.93)	(4.93)	(1.91)	(1.28)	(3.04)	(1.80)	(2.31)	(1.20)	(3.86)			
Seaman	+0.31	+1.84	−0.04	+0.62	−0.02	−0.07	−0.08	+0.00	−0.02		.63	1.42
	(1.40)	(2.86)	(0.21)	(0.90)	(0.21)	(1.17)	(3.51)	(1.53)	(1.48)			
Engineering & Hull	+0.09	+1.21	−0.02	+0.95	+0.28	−0.02	−0.03	−0.00	−0.04	+0.10	.81	1.62
	(.43)	(1.86)	(.10)	(1.32)	(2.47)	(.34)	(1.39)	(1.64)	(3.14)	(3.79)		
Construction	−0.83	−0.47	−0.03	+2.78	+0.18	+0.21	−0.20	+0.01	−0.03		.69	1.12
	(2.64)	(0.52)	(0.10)	(2.88)	(1.61)	(2.31)	(6.46)	(3.59)	(1.76)			
Aviation	−0.00	+3.69	−0.17	+1.47	+0.28	−0.03	−0.04	−0.00	−0.03		.73	1.16
	(0.00)	(5.06)	(0.84)	(1.91)	(3.08)	(0.49)	(1.62)	(1.54)	(2.61)			
Medical & Dental	+0.13	+4.05	−0.45	+0.65	−0.45	−0.30	−0.05	−0.01	−0.05		.51	0.94
	(0.41)	(4.31)	(1.73)	(0.65)	(1.73)	(0.30)	(1.54)	(1.32)	(2.90)			

Significance: For 30 or more degrees of freedom — 90% level: $t \geqslant 1.65$
— 95% level: $t \geqslant 1.96$

entrance and/or elapsed time between the decision and act of enlistment. DRFT18 was included to indicate the proportion of Navy eligibles who may have enlisted under pressure of the Draft. Presumably, draft-motivated enlistees were less prone to reenlist than their counterparts who selected service in the Navy without otherwise being compelled by the threat of induction. Except in the case of Construction, DRFT18 coefficients were negative, though very small and significant for only four occupations. Draft motivation among Navy enlistees does not appear to have been, or promise to be of major importance for first term retention.

The Viet Nam War dummy (1968-1972, all inclusive) displays unexpected differences in sign among the five categories for which it is significant. Intuitively, a negative coefficient makes sense. War is dangerous and military service in time of war is by definition a hazardous vocation. Alternatively, more challenging assignments, greater rates of advancement, and perhaps also a heightened sense of purpose may have caused the positive WAR coefficients for Ordnance and Electronics groups.

It is important to note that the WAR dummy variable was substantially more effective than an alternative which measured all-service casualty counts. While casualties were present as early as 1961, the figures increased dramatically in the late 1960's, roughly in concert with the anti-War movement. It may be the effect of that movement upon attitudes toward the military, rather than the war itself, which the WAR dummy variable is capturing.

Of central importance are the three unemployment variables in Table 2 which define national labor market activity at the time of enlistment and reenlistment. AUR is the national aggregate unemployment rate and is representative of the availability of private sector employment opportunities and of the difficulty and uncertainty associated with finding employment. As anticipated, AUR is significant, with large positive coefficients for seven of the nine occupations. Post-recession recoveries typically entail 3 to 4 points reduction in unemployment over 2 to 3 years. Taken literally, the AUR coefficients suggest these recoveries may precipitate reductions of 15 to 25 points in reenlistment rates.

As was generally true for tests of all variables in the equation, no exponential, logarithmic or other specification of AUR proved as effective as the untransformed variable. Experimentation with polynomial distributed lag functions was unproductive. No industry or occupation unemployment rates were as effective explaining reenlistment as aggregate AUR. Using national data, correlations among these various rates of unemployment are in the high 90 percentiles. Only local or regional statistics will show significantly different cyclical phasing for the different market strata to which different occupational groups might be sensitive.

The purpose of the ARAUR variable is to provide the equation with a measure of dynamics. ARAUR is the average quarterly rate of change in AUR calculated over the previous six quarters, the period determined experimentally to be the most effective.* The more rapidly the unemployment rate is changing, the less likely the individual is to perceive or believe what is happening. For a given unemployment rate, the more rapidly the rate has changed to assume its current value, the higher (for AUR falling) or lower (for AUR rising) the reenlistment rate. This is thought to be the reason for the negative ARAUR coefficients.

At least theoretically, an individual's reenlistment decision should be based upon his expectations of future private sector economic conditions. His sense of what he can earn in the private sector in both the near and distant future will be important. Short-term private sector earnings are his "opportunity cost," income he must forgo for Navy training and job experience.

*Experimentation was deemed appropriate because of uncertainty regarding the enlistee's time horizon.

Long-term prospects in the private sector will characterize the rate of return he calculates for his investment in the Navy. In general, his expectations depend upon the current condition of the economy, plus the trend of developments in the economy which he will interpret, perhaps simply extrapolate into the future. Comparing two periods in which the unemployment rate is the same, tentative reenlistees should be relatively optimistic or pessimistic about their prospects in the private sector depending on whether unemployment has been falling or rising. A priori, dynamics were expected to be a significant factor for most, if not all occupations.

Two major difficulties have probably inhibited the effectiveness of ARAUR which was significant in only three instances. One is that all such variables combine notions of speed with direction. Merging these two aspects of dynamics may be confusing if they solicit altogether different reactions from the tentative reenlistee. The second problem with dynamics is that the importance of information about trends diminishes at very high and very low values of AUR. When unemployment nears its high value in the individual's memory, its probability of falling in the future increases dramatically. Likewise, very low rates of unemployment will be expected to rise simply as an exponential function of the period over which they have persisted. Dynamics are probably most informative when unemployment rates are in their moderate range, when the future is more in doubt.

Perhaps an overriding consideration is that enlistees' information about economic conditions is derived primarily from communications from their points of origin and duty stations. These decidedly local data describe economic conditions and relevant industry and occupational market activities in the community and region in which the tentative reenlistee will consider settling. No one actually obtains employment in the national economy. National economic conditions, to which AUR refers, are in fact often unrelated to the level and dynamics of an individual area economy, a point which must be kept in mind when interpreting the significance of all the unemployment and relative wage variables.

It is interesting to note that the unemployment rate for 20-24 year olds was noticeably less effective than the aggregate rate of unemployment (AUR) which encompasses all age groups. One explanation is that data on the 20-24 year old cohort is characterized by labor market conditions in many of the major metropolitan areas, including those with large low income populations, while the Navy traditionally draws from more rural and less metropolitan areas with proportionately greater numbers of lower-middle and middle income families.

The significance of the unemployment variable, AUR13, should be especially sensitive to discrepancies between composite national and relevant local economic indicators. By the time his first term is nearing expiration, the individual is relatively distant from his home environment and even somewhat isolated from his duty station. At the outset of his term during the period to which AUR13 refers, the individual has just left his home economy. He is as knowledgeable about that local economy as he will ever be. Ideally, the equations should reference local economic conditions at the time of enlistment, rather than AUR13 which is a national unemployment rate.

As shown in Table 2, AUR13—unemployment lagged 13 periods prior to reenlistment—is significant in four instances. Most importantly, its coefficients are positive. AUR13 is believed to have bearing on the enlistee's motivation and propensity to reenlist in two respects. Unemployment (economic conditions) about the time of enlistment establishes the climate in the context of which individuals, of significantly different purpose and motivation, decide to enlist. Alternatively, unemployment about the time of enlistment describes the background against which enlistees make definite, though tentative, career decisions very early during their

first terms, decisions which are either consistent or inconsistent with reenlistment three or four years later.

Before experimentation was undertaken with "motivational" unemployment rates such as AUR13, two hypotheses were formulated to anticipate their performance in the equation. From other studies in which the authors were engaged, preliminary evidence had been developed that enlistees are either job or training oriented and will react accordingly in different ways to changes in the economy. The latter group is less inclined to reenlist. Their tendency is to view the Navy as a paid vocational college where they can effect increases in their human capital in preparation for returning to the private sector. Job oriented individuals who are more concerned with immediate employment do not see the same private sector alternatives, and are more likely to be attracted to the Navy for its long-term career potential.

Depending upon their orientation, these groups are believed to react to the Navy in opposite directions in response to economic fluctuations. The training oriented group is more prone to enlist when the economy is strong, attracted by advertised private sector positions which require skills and experience readily available from the Navy. More importantly, the composition of enlistees changes when the economy is strong because the job oriented population need not rely as heavily upon the Navy as an alternative employer. Job oriented persons will favor enlistment when the unemployment rate is high, while sophisticated training oriented types should have greater success protecting their employment status and income. The latter are more employable at enlistment and four years later when their first term is completed. When unemployment is high, the enlisted population is more job oriented and will be characterized by a higher first term reenlistment rate, suggesting a positive, significant coefficient for unemployment rates just prior to enlistment.

Interestingly, experimentation established the clear superiority of AUR13 — six to nine months after enlistment — over any lagged unemployment variable going back to, or prior to the period of enlistment. This association with unemployment in the third quarter after enlistment is supportive of a second hypothesis. Early into his first term, the typical enlistee is forming opinions about the Navy and is making conscious career decisions regarding training and the degree of his commitment to potentially long-term service. It is at this time that conditions in the private sector (probably in his home town) are taken into account.* If his experiences have been generally negative, he may look more favorably upon the private sector. If the economy is healthy, he may decide not to work to enhance his status in the Navy beyond what he considers necessary to maximize his success upon return to the private sector. Having never fully committed himself to the Navy, the enlistee never seriously considers extending his service through a second term.

In contrast to unemployment variables to which policymakers can only react, RW (relative wages) is a parameter over which the Navy has direct control. (The relative wage variable was calculated with reference to E-4 base pay, excluding indirect benefits and bonuses, lump sum or installment. RW is the ratio of E-4 base pay to the earnings of private nonagricultural, nonsupervisory production workers.) Relative wages were significant for all but Seaman and Construction personnel, with predictably positive, though small coefficients. The implication is that relative wages, exclusive of bonuses or benefits, are ineffective as a policy variable to cause independent changes in reenlistment or to combat the effects of an improving economy.

*During their first six months of service, recruits have two weeks leave, an early opportunity to evaluate their enlistment decisions at home in the company of family and friends, and with an unobstructed view of local private sector labor market conditions.

Even more troublesome regarding the efficacy of relative wages as a policy instrument is that the meaning of the RW variable is suspect. Values for RW trace out a low grade exponential curve which, especially in recent years, parallels quality of life improvements which have been implemented for enlisted personnel. RW is the only variable in the equation which follows this general form and may simply be serving as a proxy for other factors favorable to reenlistment which have not, and probably could not be captured by the equations.*

IV. EARLY AND RECENT SAMPLES

The dynamics of the national economy changed after 1970. Following almost a decade during which the economy experienced continued improvement, the 1970's brought two sharp recessions. These fluctuations occurred in the context, and perhaps to some extent as a result of a barrage of new socio-political and technological phenomena and events which grew out of the Viet Nam Conflict and coincided with the maturing of post-World War II baby-boom labor. Even the legendary work ethic which has supposedly sustained the character of the American economy since its inception began to suffer a noticeable loss of popularity.

It can be assumed that, against this background of complex and rapid change, a new business cycle and labor force mentality have emerged from the late 1960's. Attitudes toward the military cannot have been unaffected. It follows that the relationships between reenlistment and its determinants, especially unemployment and relative wages, may have also been altered and differ now from what they were a decade ago. More than likely, a single model describing the complete twenty year time span will not be appropriate for projecting reenlistment behavior in the near future. Reenlistment over the next five to 10 years will probably be more consistent with its history during the very recent past, beginning in the late 1960's, early 1970's.

To test for the existence of unique recent period relationships, the 80 quarter reenlistment time series was split into two 10 year samples, 1/1958-4/1967 and 1/1968-4/1977.† The results of the earlier sample are shown in Table 3. Compared to the equations based on all 80 observations, these early sample equations are obviously less effective. In addition to the seasonal dummy, the equations are dominated by the unemployment rate variables, especially current period AUR. Neither the Draft variable nor RW (relative wages) was significant for a single occupational category. With the exception of the Aviation and Medical categories, the generally low R^2 statistics indicate that the early sample equations are improperly, or more likely underspecified. DRFT18 and RW might be significant in the context of a more explicit, more complete model.‡ The point remains, however, that the same factors (with the exception of the WAR variable which was not defined prior to 1968) which explain reenlistment to a reasonable degree of effectiveness over twenty years have failed to repeat that achievement for a shorter time frame.

The results of the later sample are shown in Table 4. Most notable, as compared to the earlier sample, are the R^2 and constant terms.** The R^2 statistics are impressive, and the

*ARATE, the average rate of first term eligibles, exhibits a gradual, continual increase over time. ARATE was introduced to recognize differences in attitudes and earnings among enlistees which would be a function of levels of achievement. In fact, ARATE may be driven by reenlistment rates as a result of changes in promotion policies.

†Analysis of correlation matrices indicated that multicollinearity was not a problem in either of these sample periods.

‡The insignificance of the Draft variable (DRFT18) in the early equations probably derives from the peacetime period for which the variable was relevant. DRFT18 describes motivation at the time of enlistment, and therefore measures peacetime levels of inductees, 1953-1963.

**The WAR variable, absent in the earlier sample, was not responsible for the generally remarkable performance of the recent sample equations. This conclusion was substantiated by tests which estimated equations for the recent sample without the WAR variable present. Changes in results were nominal, with only the Durbin-Watson statistics showing any appreciable deterioration.

TABLE 3. *Early Ten Year Sample*
Determinants of Navy Reenlistment (Quarterly: 1/58-4/67)
Coefficients (t-Statistics) For All Variables

	Independent Variables										
	C	AUR	ARAUR	AUR13	RW	ARATE	WAR	DRFT18	S3	R^2	D-W
Deck	+0.58	+2.65	-0.25	+0.56	+0.35	-0.16	—	-0.01	-0.04	.36	1.70
	(2.00)	(1.94)	(0.89)	(0.71)	(1.30)	(2.13)		(1.02)	(2.79)		
Ordnance	+0.59	-1.18	-0.09	+1.45	-0.18	-0.06	—	-0.01	-0.05	.48	2.13
	(1.86)	(0.80)	(0.28)	(1.68)	(0.61)	(0.75)		(1.53)	(3.26)		
Electronics	-0.56	-1.10	-0.01	+3.01	+0.42	+0.15	—	-0.01	-0.07	.52	1.81
& Prec. Equip.	(1.29)	(0.54)	(0.02)	(2.52)	(1.03)	(1.29)		(1.15)	(3.01)		
Administration	+0.54	+5.37	-0.71	-0.24	+0.33	-0.16	—	-0.00	-0.06	.59	2.17
	(1.81)	(3.80)	(2.45)	(0.30)	(1.17)	(2.03)		(0.64)	(3.84)		
Seaman	+0.20	+4.26	-0.32	-0.28	+0.19	-0.08	—	-0.00	-0.02	.57	1.97
	(0.76)	(3.43)	(1.26)	(0.38)	(0.78)	(1.20)		(0.47)	(1.63)		
Engineering & Hull	+0.25	+2.21	-0.13	+0.18	+0.19	-0.06	—	-0.00	-0.03	.36	2.17
	(1.08)	(2.06)	(0.58)	(0.29)	(0.87)	(0.93)		(0.85)	(2.90)		
Construction	-0.25	+3.31	-0.50	+1.17	-0.20	+0.07	—	+0.00	-0.04	.51	1.52
	(0.82)	(2.34)	(1.72)	(1.41)	(0.70)	(0.92)		(0.60)	(2.61)		
Aviation	+0.08	+5.27	-0.14	-0.24	-0.02	-0.03	—	-0.00	-0.04	.74	2.22
	(-.34)	(4.53)	(0.57)	(0.35)	(0.08)	(0.45)		(1.07)	(3.11)		
Medical	+0.22	+7.69	-0.84	-1.48	-0.27	-0.05	—	-0.00	-0.06	.77	1.91
& Dental	(0.70)	(5.16)	(2.75)	(1.71)	(0.90)	(0.55)		(0.51)	(4.01)		

Significance: For 30 or more degrees of freedom — 90% level: $t \geqslant 1.65$
— 95% level: $t \geqslant 1.96$

TABLE 4. *Recent Ten Year Sample*
Determinants Of Navy Reenlistment (Quarterly: 1/68-4/77)
Coefficients (t-Statistics) For All Variables

Occupations	Independent Variables										
	C	AUR	ARAUR	AUR13	RW	ARATE	WAR	DRFT18	S3	R^2	D-W
Deck	−1.12	+5.25	−1.09	+8.64	+0.02	+0.16	+0.07	−0.00	−0.03	.88	1.98
	(1.88)	(4.56)	(3.49)	(5.08)	(0.08)	(1.13)	(1.85)	(1.59)	(1.75)		
Ordnance	−2.58	+8.26	−1.44	+6.30	−0.05	+0.53	+0.11	−0.01	−0.05	.78	1.42
	(2.65)	(4.37)	(2.82)	(2.26)	(0.14)	(2.32)	(1.70)	(1.39)	(1.81)		
Electronics & Prec. Equip.	−1.42	+7.16	+0.61	+3.82	+0.61	+0.28	−0.01	−0.01	−0.02	.73	1.01
	(0.86)	(2.09)	(0.71)	(0.82)	(1.01)	(0.73)	(0.10)	(0.50)	(0.39)		
Administration	−1.20	+2.80	−0.53	+6.96	+0.01	+0.24	−0.02	−0.00	−0.03	.89	2.03
	(2.35)	(2.82)	(2.00)	(4.77)	(0.08)	(1.98)	(0.52)	(1.30)	(2.08)		
Seaman	−1.38	−0.37	+0.27	+6.41	+0.41	+0.23	+0.03	−0.00	−0.00	.81	1.30
	(6.41)	(0.41)	(1.11)	(4.80)	(2.41)	(2.11)	(1.03)	(1.57)	(0.33)		
Engineering & Hull	−1.56	+1.23	−0.24	+8.03	+0.33	+0.27	−0.01	−0.07	−0.02	.89	1.41
	(2.86)	(1.10)	(0.78)	(4.88)	(1.57)	(2.08)	(0.22)	(1.93)	(1.31)		
Construction	−1.68	−5.12	+0.71	+6.44	+0.41	+0.40	−0.22	+0.01	−0.00	.89	1.72
	(2.45)	(3.85)	(1.98)	(3.29)	(1.62)	(2.51)	(4.90)	(1.51)	(1.51)		
Aviation	−1.34	+1.29	−0.37	+8.26	+0.33	+0.22	+0.02	−0.01	−0.02	.91	2.12
	(2.75)	(1.90)	(1.46)	(5.92)	(1.85)	(1.95)	(0.65)	(2.56)	(1.44)		
Medical & Dental	−1.68	−1.22	+0.28	+8.22	+0.80	+0.22	−0.08	−0.01	−0.01	.79	1.54
	(2.76)	(1.04)	(0.87)	(4.74)	(3.56)	(1.90)	(1.97)	(2.18)	(0.33)		

Significance: For 30 or more degrees of freedom — 90% level: $t \geq 1.65$
— 95% level: $t \geq 1.96$

constants generally suggestive of equations which describe different relationships than those captured by the early and total sample equations. Certainly, the single most important reason for the superior performance of the recent sample equations is the overall strength of the unemployment indicators, AUR, ARAUR and AUR13.* Assuming that attitudes toward military service have been altered, it is likely that they have changed to favor a heightened sensitivity to economic fluctuations. More and more recruits may be viewing military service strictly as a vocational decision. If they are training oriented, they watch the economy to discern when they can most effectively capitalize on the increases in their human capital which the Navy is providing them. If they are job oriented and see the Navy as a fall-back alternative to private sector unemployment, they may remain in the Navy only until they detect better opportunities on the outside. Either way, job or training oriented, the enlistee will do his best to keep in touch with economic conditions, more now than in the 1960's when economic motivations were less influential relative to social and personal psychological factors.

*The negative coefficient for AUR in the Construction equation (Table 4) may indicate that Navy Construction personnel identify with the public works (infrastructure) component of the construction industry. Public works construction is sometimes undertaken as part of counter-cyclical employment programs, and must precede or follow residential and commercial/industrial development. As such, public works employment may experience cycles approaching 180 degrees out of phase with other construction activities which are more closely associated with movements in the national unemployment rate. This assumption about Navy construction personnel is not necessarily inconsistent with the typically positive coefficient for AUR13. Six to nine months into his first term, the new recruit may not yet consider himself affiliated with the construction industry.

RW is significant and positive for only three occupational categories, and of minor impact in these instances. Redefining relative wages based upon Regular Military Compensation did nothing to increase the number of significant cases, nor did it affect significant RW coefficients substantially. Taking into account housing and subsistence allowances and the associated tax advantages produced results which were no more impressive.

V. FINDINGS AND IMPLICATIONS

For the analysis, conceptualization and timing of programs designed to anticipate or control reenlistment, of the three models (early or recent samples, or all 80 observations) the recent sample equations would seem to be most pertinent for two reasons. First, economic conditions in the near future are more likely to resemble the early 1970's than the 1960's. Those social, political and economic phenomena which have precipitated the new dynamics of recent fluctuations are likely to persist. Second, contemporary attitudes toward military service are part of a generally irreversible evolution of mores and traditions. Intuitively, changes in attitudes should involve an increasingly heavy emphasis by recruits upon the vocational aspects of service. Sensitivities to, and knowledge of economic conditions should be increasing, perhaps as indicated by the performance of unemployment rate variables, particularly AUR13, in the recent sample equations. Quite probably, it is these relationships which are responsible for the effectiveness of equations based on the entire twenty year time series.

Assuming that neither war nor the Draft are likely to repeat themselves in the near future, the only recurrent determinants of reenlistment — other than subjective factors not captured by the equations — are unemployment (economic conditions) and relative wages. Relative wages, now that parity between military and private sector compensation is guaranteed by law, are effectively constant, excluding the payment of bonuses. That leaves the economy, and perhaps bonuses also, as the dominant influences which will characterize reenlistment behavior in the near future. (Time series data on which the study was based precluded incorporating lump sum or installment bonuses into either the RW variable or elsewhere in the equations.)

As Table 5 indicates, with reference only to unemployment rate variables, it is possible to explain from 51 to 86 percent of the variation in reenlistment rates over the past ten years.* The significance of the unemployment variables is particulary impressive considering:

1. The equations include no quality of life indicators representative of improvements which have occurred to enlisted working conditions.

2. No sampling has occurred to separate individuals by sex, ethnicity, family status or mental group.

3. The unemployment rate data being used is national and aggregate. No direct references have been made to local economic conditions in the enlistee's home area or duty station.

Without question, reenlistment rates are highly sensitive to economic conditions at reenlistment and enlistment, represented by unemployment rates as indicators of the availability and difficulty of securing private sector employment. The effect of current unemployment is quite strong, but the effects of unemployment about the time of enlistment, AUR13, is especially pronounced.

*Some R^2 statistics are undoubtedly biased upwards due to the presence of positive serial correlation.

TABLE 5. *Recent Ten Year Sample*
Determinants of Navy Reenlistment (Quarterly 1/68-4/77)
Coefficients (t-Statistics) for Significant Variables

Occupations	Independent Variables					
	C	AUR	ARAUR	AUR13	R^2	D-W
Deck	-0.30 (6.80)	+4.77 (7.41)	-0.97 (4.80)	+6.28 (6.07)	.83	1.54
Ordnance		+7.89 (7.27)	-1.23 (3.61)		.68	1.19
Electronics & Prec. Equip.		+10.44 (6.39)			.69	0.83
Administration	-0.26 (6.92)	+4.02 (7.14)	-0.78 (4.41)	+6.69 (7.41)	.85	1.92
Seaman	-0.22 (5.85)	+1.79 (3.18)		+4.77 (5.27)	.67	1.12
Engineering & Hull	-0.30 (6.70)	+3.88 (5.84)	-0.78 (3.72)	+7.83 (7.34)	.82	1.36
Construction	-0.33 (3.70)			+10.27 (4.82)	.51	0.43
Aviation	-0.32 (8.16)	+3.95 (6.84)	-0.81 (4.48)	+7.40 (7.98)	.86	1.70
Medical & Dental	-0.15 (2.75)	+2.13 (2.63)		+5.46 (4.19)	.56	1.03

Note: Coefficients and t-statistics are omitted for variables not significant at the 90% level.
For 30 or more degrees of freedom—90% level: $t \geqslant 1.65$; 95% level: $t \geqslant 1.96$.

The effects of AUR and AUR13 combined can be devastating for reenlistment. Compare two "classes" of recruits, one enlisting and coming up for reenlistment during peaks in the economy, the other during low points. Assuming these peaks and troughs are separated by only two percent, six to eight percent unemployment for example, the total difference in reenlistment rates between the two groups could be as high as 27 to 29 percent for Deck and Ordnance, as low as three percent for Construction.

Striking by comparison is the poor performance of relative wages, the variable RW. Statistically significant for only three occupations, its effect in those cases is nominal, especially considering the magnitude of the unemployment rate coefficients. Calculating the rates of substitution between unemployment and relative wages, the latter appears ineffective as a means of protecting reenlistment rates from a healthy, vigorous private sector.

Comparison of equations suggests that the Navy's enlisted workforce is not a homogeneous mass for which precisely the same reenlistment programs would be appropriate. Differences in reactions to the economy and relative wages among the nine major occupational groups which were studied are evident and of significant order of magnitude. The policy implication is that different occupations should be treated differently to effect comparable degrees of control over their reenlistment rates.

Unemployment rates, current and at the time of enlistment, are the principal determinants of reenlistment. The effects of changes in current economic conditions (AUR) are substantial, in the neighborhood of from 4 to 5 percentage points change in reenlistment rates for every 1 point change in unemployment. Moreover, the significance and power of the AUR13 variable is particularly important for the nature and timing of programs designed to increase reenlistment. It is apparent that the propensity to reenlist is to a great extent determined very early in the first term, at or about the time of enlistment. The importance of AUR13 has profound implications for the incidence of recruiting expenditures and early enlistment counseling programs. Finally, although for some occupations they may be a significant determinant reenlistment, military relative to private sector rates of compensation are generally powerless to compensate for the effects of economic fluctuations.

To the extent that bonuses paid in installments over long periods are viewed as regular wages, they may be ineffective as reenlistment incentives. Other than their importance for enlistment, relative wages can probably be allowed to deteriorate when economic conditions favor rising reenlistment rates.* More importantly, the order of magnitude of increases in military compensation — excluding bonuses — necessary to protect reenlistment rates when economic conditions are driving those rates down is almost assuredly financially and politically unacceptable. No evidence was discovered indicating that reenlistment has been affected by variations in benefits for housing, subsistence, or tax advantages.

If relative wages are not effective as a program instrument to control reenlistment, the Navy has only two alternatives: lump-sum incentive grants which are more impressive, at least more visible than wage increases (or installment bonuses) and probably less expensive for a given impact; and procedural changes affecting the pace of promotions and intra-Navy job and occupational mobility so as to bring the career patterns of enlisted personnel more in line with those of their private sector counterparts. The latter may be especially important in light of the orientation toward employment opportunities and lack of concern for direct and indirect monetary compensation which this study has identified.

Differences between Navy and private sector rates of achievement and mobility — vertical and horizontal — may produce a degree of discontentment and sense of falling behind the private sector. This problem has potentially serious ramifications for first and even second term reenlistment, and is clearly a policy issue deserving immediate study. Unemployment rates are factors to which the Navy can react or for which it can prepare, but not control. Since relative wages are now basically constant and if they are generally powerless to counteract economic stimuli, substantial lump sum bonus programs and improvements in career development paths may be necessary to counteract or nullify the link between business cycles and reenlistment.

VI. RESERVATIONS AND LIMITATIONS

There are the standard set of reservations associated with regression analysis and all methods of statistical inference involving correlation. Specifically with respect to the equations discussed above, there are a few problems. The signs of some coefficients are troublesome.

*There is evidence that the major proportion of tentative enlistees are either oblivious or insensitive to rates of military compensation. At most, those who do not enlist may be somewhat put off by a general impression that military pay is relatively low compared to the private sector. See for example, D. Grissmer, et al., "An Econometric Analysis of Volunteer Enlistments by Service and Cost Effectiveness Comparison of Service Incentive Programs," OAD-CR-66, General Research Corporation, October 1974.

The Durbin-Watson statistics are generally indicative of serial correlation among error terms, more severe for some occupations than others. As noted above, the RW variable is suspect. Its mildly exponential behavior is coincidental with a number of other phenomena which might have also affected reenlistment. Despite these difficulties, the equations do well and are reliable for what they convey about the direction and order of magnitude of the effects of unemployment upon reenlistment.

The principal limitations to direct policy application of these findings derive from the simplicity of the time series data on which they are based. Without specific ratings data, reenlistment bonuses could not be taken into account. Available time series data prohibit effecting any controls for personal or socio-demographic considerations, notably ethnicity, sex, family status and especially mental group. The time series analysis above fails to pinpoint who is leaving and to explain precisely why they leave.

Equally important, the data have prevented any reference to labor market conditions in either the enlistee's duty station or home economy where he might consider settling. Local economies differ substantially in the way they experience phases of any national cycle, differences which a complete analysis of reenlistment behavior must take into account. To a great extent, compensation for these disadvantages of time series analysis could be accomplished via supplementary cross-sectional analysis of area reenlistment rates using the Enlisted Master File or some related set of records.

ACKNOWLEDGMENTS

The authors gratefully acknowledge the technical assistance of Ms. Deborah Coffin without whom this paper would have suffered considerable loss of substance and detail, and the valuable criticisms and comments of Drs. Alfred Rhode and John Martin of Information Spectrum, Inc., Mr. Irwin Schiff and LCDR Kevin Delaney of OP-964D, and Mr. Samuel Kleinman of the Center for Naval Analyses.

BIBLIOGRAPHY

[1] Albrecht, M., "A Discussion of Some Applications of Human Captial Theory to Military Manpower Issues," P-5727, RAND, September 1976.

[2] Bryan, J. and A. Singer, "Prediction of Reenlistment Using Regression Estimation of Event Probabilities," Research Contribution No. 13, Center for Naval Analyses, October 1965.

[3] Cooper, R., "The All-Volunteer Force: Five Years Later," P-6051, RAND, December 1977.

[4] Enns, J., "Effect of the Variable Reenlistment Bonus on Reenlistment Rates: Empirical Results for FY-71," R-1502-ARPA, RAND, June 1975.

[5] Grissmer, D., et al., "An Econometric Analysis of Volunteer Enlistments by Service and Cost Effectiveness Comparison of Service Incentive Programs," OAD-CR-66, General Research Corporation, October 1974.

[6] Haber, S. and C. Stewart, Jr., "The Responsiveness of Reenlistment to Changes in Navy Compensation," TR-1254, George Washington University, May 1975.

[7] Lindsay, W., Jr. and B. Causey, "A Statistical Model for the Prediction of Reenlistment," TP-342, Research Analysis Corporation, March 1969.

[8] Lindsay, W., et al., "Simple Regression Models for Estimating Future Enlistment and Reenlistment in Army Manpower Planning," TP-402, Research Analysis Corporation," September 1970.

[9] Lockman, R., et al., "Motivational Factors in Accession and Retention Behavior," Research Contribution 201, Center for Naval Analyses, January 1972.

[10] Massell, A., "An Imputation Method for Estimating Civilian Opportunities Available to Military Enlisted Men," R-1565-ARPA, RAND, July 1975.
[11] Massell, A., "Reservation Wages and Military Reenlistments," P-55336, RAND, February 1976.
[12] Nelson, G., "An Economic Analysis of First Term Reenlistments in the Army," P-647, Institute for Defense Analyses, June 1970.
[13] Quigley, J. and R. Wilburn, "An Economic Analysis of First Term Reenlistment in the Air Force," AFPDPL-PR-69-017, Personnel Research and Analysis Division, Directorate of Personnel Planning, USAF, September 1969.
[14] Young & Rubicam, Inc., "Naval Retention: A Problem of Empathy," May 1970.

INDEX TO VOLUME 26

☆U.S. GOVERNMENT PRINTING OFFICE: 1979 O—305-622

INFORMATION FOR CONTRIBUTORS

The NAVAL RESEARCH LOGISTICS QUARTERLY is devoted to the dissemination of scientific information in logistics and will publish research and expository papers, including those in certain areas of mathematics, statistics, and economics, relevant to the over-all effort to improve the efficiency and effectiveness of logistics operations.

Manuscripts and other items for publication should be sent to The Managing Editor, NAVAL RESEARCH LOGISTICS QUARTERLY, Office of Naval Research, Arlington, Va. 22217. Each manuscript which is considered to be suitable material for the QUARTERLY is sent to one or more referees.

Manuscripts submitted for publication should be typewritten, double-spaced, and the author should retain a copy. Refereeing may be expedited if an extra copy of the manuscript is submitted with the original.

A short abstract (not over 400 words) should accompany each manuscript. This will appear at the head of the published paper in the QUARTERLY.

There is no authorization for compensation to authors for papers which have been accepted for publication. Authors will receive 250 reprints of their published papers.

Readers are invited to submit to the Managing Editor items of general interest in the field of logistics, for possible publication in the NEWS AND MEMORANDA or NOTES sections of the QUARTERLY.

NAVAL RESEARCH
LOGISTICS
QUARTERLY

DECEMBER 1979
VOL. 26, NO. 4
NAVSO P-1278

CONTENTS

OFFICE OF NAVAL RESEARCH
Arlington, Va. 22217